UNITED STATES AIR FORCE
RESEARCH LABORATORY

TELEROBOTIC CONTROL ARCHITECTURE
INCLUDING FORCE-REFLECTION (U)

Mark A. Murphy
Robert L. Williams, III

OHIO UNIVERSITY
ATHENS OH 45701

JUNE 1998

FINAL REPORT FOR THE PERIOD 16 JUNE 1997 TO 31 MAY 1998

20030310 063

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2002-0186

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

MARIS M. VIKMANIS
Chief, Crew System Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1998	3. REPORT TYPE AND DATES COVERED FINAL REPORT, 16 June 1997 to 31 May 1998

4. TITLE AND SUBTITLE

Telerobotic Control Architecture Including Force-Reflection (U)

6. AUTHOR(S)

* Mark A. Murphy
* Robert L. Williams, III

5. FUNDING NUMBERS

C: F41624-95-C-6014
PE: 62202F
PR: 7184
TA: 08
WU: 69

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ohio University
Athens OH 45701

8. PERFORMING ORGANIZATION

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING

AFRL-HE-WP-TR-2002-0186

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes the implementation of a telerobotic control architecture to manipulate a standard six-degree-of-freedom robot via a unique seven-degree-of-freedom force-reflecting exoskeleton which is located in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base. This is the first time that the robot and exoskeleton have been interfaced. The novel Naturally-Transitioning Rate-to-Force Controller is included in the implementation. Background for the control architecture and modes of operation are presented as well as the specific system description and operating procedures. Peg-insertion experiments were conducted to compare the performance of rate control, naturally-transforming rate-to-force control, and naturally-transforming rate-to-force control with force reflection. Task completion time and manipulator work due to contract forces moment through the Cartesian displacements were the basis for comparison. The control architecture is completely implemented. Experimental results displayed no clear differences among the control modes; this indicates that a reduction in system time delays and more precise gain tuning are needed.

14. SUBJECT TERMS

Force Reflection, Teleoperation, telerobotic

15. NUMBER OF PAGES
102

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89) Prescribed by ANSI Std Z-39-18
298-102 COMPUTER GENERATED

This Page Intentionally Left Blank

PREFACE

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1 INTRODUCTION

This report describes the implementation of a telerobotic control architecture to manipulate a standard six degree of freedom industrial robot via a unique seven degree of freedom force-reflecting exoskeleton which is located in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base. This is the first time that the robot and exoskeleton have been interfaced.

The novel Naturally-Transitioning Rate-to-Force Controller is included in the implementation. Background for the control architecture and modes of operation are presented as well as the specific system description and operating procedures.

Peg-insertion experiments were conducted to compare the performance of rate control, Naturally-Transitioning Rate-to-Force Control, and Naturally-Transitioning Rate-to-Force Control with force reflection. Task completion time and manipulator work due to contact forces and moments through Cartesian displacements were the basis for comparison. The control architecture has been completely implemented. Experimental results displayed no clear differences among the three control modes; this indicates that a reduction in system time delays and more precise gain tuning are needed.

Many tasks that require the reasoning and reactions of a human can be performed by teleoperation of a robotic manipulator. Unlike most current industrial robotics applications that are automated and consist of repetitive tasks, teleoperation is best suited for tasks that require improvisation.

Teleoperation is not a new technology. The recent evolution of computers has provided the neural network required to further advance the field. Johnsen and Corliss (1971) provided an introduction to teleoperation and its early chronology.

Teleoperation allows a person to remotely perform tasks in environments that are hazardous to human health. Space, water, and underground environments have obvious teleoperation applications. Skaar and Ruoff (1994) and Tzafestas (1991) provided an overview of teleoperation for space applications. Current applications also include tasks in radioactive areas as well as fire fighting and bomb diffusing. The Air Force teleoperation applications and research goals are discussed by the Committee on Advanced Robotics for Air Force Operations, Air Force Studies Board, Commission on Engineering and Technical Systems, and National research Council (1989).

To evaluate something it must be compared to a standard. The human in the loop complicates the evaluation of teleoperation control methods since there is no standard human (Vertut & Coiffet, 1985/1986). The time required to complete a task usually does not provide sufficient information to evaluate teleoperation control methods since other factors such as safety, reliability, contact force, or strength could be more important while performing the desired task. There are many different teleoperation applications and therefore it is impossible to choose a specific control method to use in all teleoperation tasks. Each situation should be evaluated separately considering the advantages and disadvantages of each control method.

Theoretically, teleoperation of remote manipulators is greatly enhanced by using a force-reflecting input device. This force and moment haptic feedback should increase the sense of telepresence, the sense of occupying the remote or virtual environment, by

enabling the operator to feel through the force-reflecting master the forces and moments exerted by the slave manipulator on the environment. The Human Sensory Feedback (HSF) Laboratory located at Wright-Patterson Air Force Base has a world-class capability for experimentation in force-reflecting teleoperation for Air Force and NASA applications: The unique *FREFLEX* (*F*orce-*REFL*ecting *EX*oskeleton, a unique device from an Odetics, Inc. SBIR [Odetics, Inc., 1992]) force-reflecting exoskeleton master and a *MERLIN* (*M*odular *E*xpandable *R*obot *LIN*e, a common industrial robotic arm from American Robot Corporation [American Robot Corporation,1985]) industrial manipulator slave.

The HSF Lab has been involved with force-reflecting teleoperation research for more than a decade. Bryfogle (1990) presented algorithms for force-reflecting exoskeletons. Rosenberg (1992) applied virtual fixtures to improve teleoperator performance and later extends the concept to include time-delayed teleoperation (1993). Huang (1993) presented equations for FREFLEX exoskeleton inputs and the MERLIN inverse pose solution, optimized for minimal on-line computation. Dr. Repperger has been very active in force-reflection research, focusing on the operator side of teleoperation (Repperger, 1991, 1995; Repperger, Phillips, & Chelette, 1995; Repperger, Phillips, Hill, & Roark, 1996; Repperger, Scarborough, & Chelette, 1991).

A unique experimental sensor-based real-time telerobotic system including force-reflecting hand controllers (FRHCs) was implemented at NASA Langley Research Center (Williams, Harrison, & Soloway, 1996, 1997; Willshire, Harrison, Hogge, Williams, & Soloway, 1992). The control architecture discussed in this report is an extension and adaptation of that system, which includes multiple control modes and shared human and autonomous control.

The main objective of this research was to expand the capability of the HSF lab to perform teleoperation experiments by implementing a telerobotic control architecture that allows force reflection to the teleoperator and coordination of kinematically dissimilar FRHCs and slave robots. The HSF lab has possessed the hardware, MERLIN and FREFLEX, to perform these evaluations for several years but these devices were never interfaced. The implemented telerobotic control architecture includes not only the usual teleoperation control modes such as pose and rate control but also the novel Naturally-Transitioning Rate-To-Force Controller (NTRFC). The only previous implementation of the NTRFC was at NASA Langley Research Center (Williams et al., 1996) and it was purely heuristic. The HSF lab will utilize the telerobotic capabilities provided by this research to investigate sensory-rich, human-in-the-loop control and define performance measures for advanced human sensory feedback development.

The research covered in this report advanced the teleoperation research capabilities and knowledge of the HSF Lab by supplying a versatile telerobotic control architecture for use in teleoperation studies. Controlling the MERLIN via the FREFLEX and modeling the NTRFC were accomplished for the first time. Several control methods were installed and experiments to evaluate control method performance were initiated for peg-in-hole tasks. Task completion time and manipulator work due to contact forces and moments through Cartesian displacements were the basis for comparing the implemented methods of teleoperation.

This report summarizes the following research accomplishments: (a) development of control architecture for general telerobotic systems including FRHCs; (b) simulation, implementation, and evaluation of the telerobotic control architecture applied to the FREFLEX and MERLIN system; and (c) controls design, modeling, simulation, and evaluation of the novel NTRFC, which is part of the control architecture for general telerobotic systems including FRHCs. The general system that is applicable to the control architecture and the specific HSF lab system are described in Chapter 2. Chapter 3 presents the telerobotic control architecture. The NTRFC is discussed and modeled in Chapter 4. Chapter 5 summarizes the MATLAB simulation that was used to develop the control algorithms and describes the implementation of the control architecture into the HSF hardware. The experimental comparison of resolved-rate control, the NTRFC with force reflection, and the NTRFC without force reflection is presented in Chapter 6. Chapter 7 summarizes the research, emphasizes the significant conclusions, and recommends future research and system improvements.

CHAPTER 2   SYSTEM DESCRIPTION

This chapter discusses the general system characteristics used to derive the control theory and the specific system in the HSF Lab to which the theory was applied.

2.1 General System Characteristics

This report assumes the following general system characteristics:

1. One or more slave manipulators are to be controlled to accomplish various tasks.

2. One or more manipulators may be controlled by human operator (teleoperation), autonomously (robotic) or a combination (telerobotic).

3. The slave manipulators should posses at least six degrees of freedom for general spatial tasks.

4. A master device such as a joystick, hand controller, or exoskeleton with at least six degrees of freedom was used for teleoperation inputs.

Since Cartesian commands from the master are sent as Cartesian commands to the manipulator or manipulators, the master and slave need not be kinematically similar. Cartesian master to Cartesian slave control has more capability than joint to joint control. If two slave manipulators are working independently, two master devices may be used. If two slave manipulators are coupled through a common payload, a single master is sufficient. Figure 1 shows coordinate frame definitions which apply to masters and slaves.
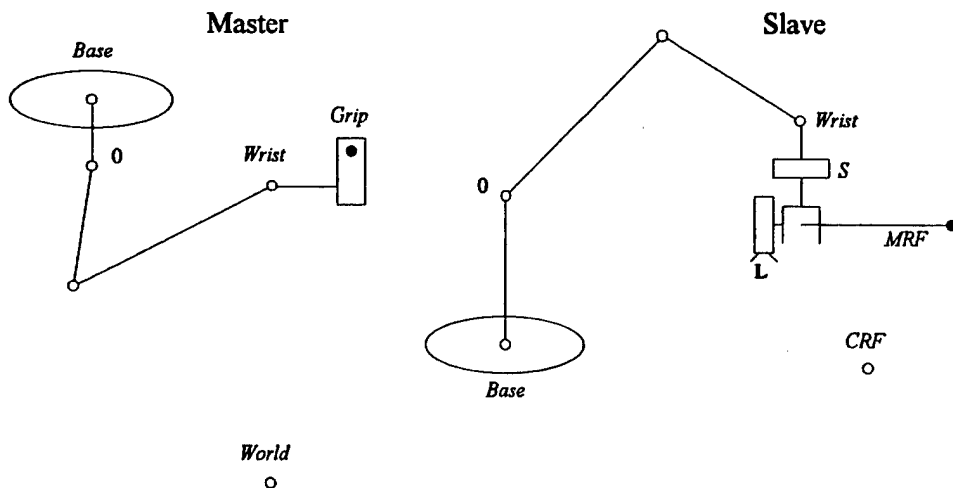
Figure 1. Master and Slave Coordinate Frames

For clarity, dextral *XYZ* Cartesian coordinate frames are represented by dots and small circles in Figure 1. The *World Frame* is an inertially-fixed reference frame for all devices. The master and slave each have separate *Base, 0,* and *Wrist Frames*. The Base Frame is attached before the first moving joint; 0 is the kinematic base frame; the Wrist Frame is attached to the last moving link at its joint.   The master and slave each have coordinate frames attached to each active joint between 0 and the wrist which is not

shown for generality and clarity. The master *Grip Frame* is centered at the human operator's hand grasp point. The slave has the following frames: The *Moving Reference Frame* (MRF) is a user-defined frame which is being controlled. The MRF can be placed anywhere as long as it is rigidly attached to the last manipulator link, such as on a grasped payload or even off the physical link. The *Control Reference Frame* (CRF) is a user-defined frame with respect to which the MRF is controlled. Cartesian velocities may be commanded in the coordinates of any frame, but all motion relates the MRF to the CRF. The *L Frame* is the camera lens which is used for machine vision, remote operator views, or both. The *S Frame* is the force and torque sensor frame. The *L* and *S* Frames are both rigidly attached to the slave Wrist Frame and MRF.

The control frames in Figure 1 are defined for generality. The CRF can be moving and the Base Frame can also be moving independently with respect to the World Frame. The MRF can be changed during tasks and is defined to facilitate task completion. For example, the MRF can be the beam node in a beam assembly task. In this case the CRF would be the target connecting node location. The inclusion of the MRF and CRF is intended to decouple the Cartesian task and the human operator from the slave manipulator. Figure 2 shows the general control flow in a force-reflecting teleoperated system. *Pose* stands for Cartesian position and orientation and *wrench* represents a Cartesian force and moment vector in Figure 2 and throughout this report. In this report a force-reflecting master will be generically referred to as a force-reflecting hand controller.
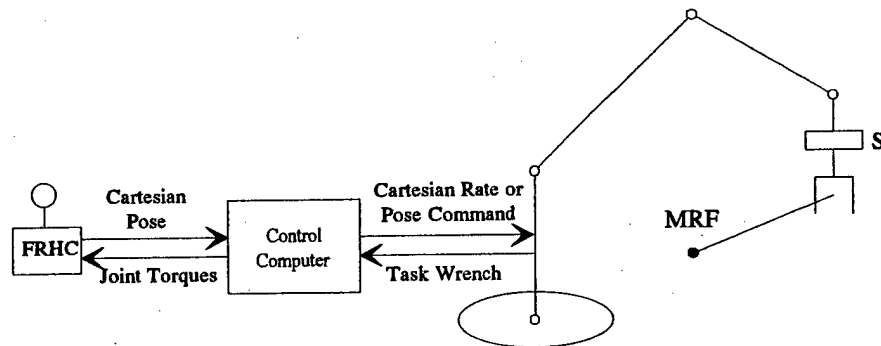
Figure 2. Force-Reflecting Teleoperated System

2.2 HSF Lab System Description

The HSF Lab has the following devices for teleoperation experimentation. The control methods of this report were implemented on these devices in simulation and hardware. A single MERLIN 6500 robot arm (American Robot Corporation, 1985) shown in Figure 3 is the slave manipulator. This six degrees of freedom spatial device consists of six revolute joint axes in series. A second MERLIN may be available in the future to implement dual arm control. The master is the seven degrees of freedom, seven revolute joint FREFLEX shown in Figure 4. Chimera 3.2, which is a real-time operating system for reconfigurable sensor-based control systems developed by the Advanced Manipulators Laboratory, The Robotics Institute, and the Department of Electrical and

Computer Engineering at Carnegie Mellon University (Ingimarson, Stewart, & Khosla, 1995) is installed on a Sun SPARCstation and acts as the interface between the FREFLEX and the MERLIN. Two JR[3] Universal Force-Moment Sensor Systems (JR[3], Inc., 1988) are available. One of the sensors is mounted on the wrist of the FREFLEX and the other is attached at the wrist of the MERLIN. A taskboard for teleoperation studies is also available.
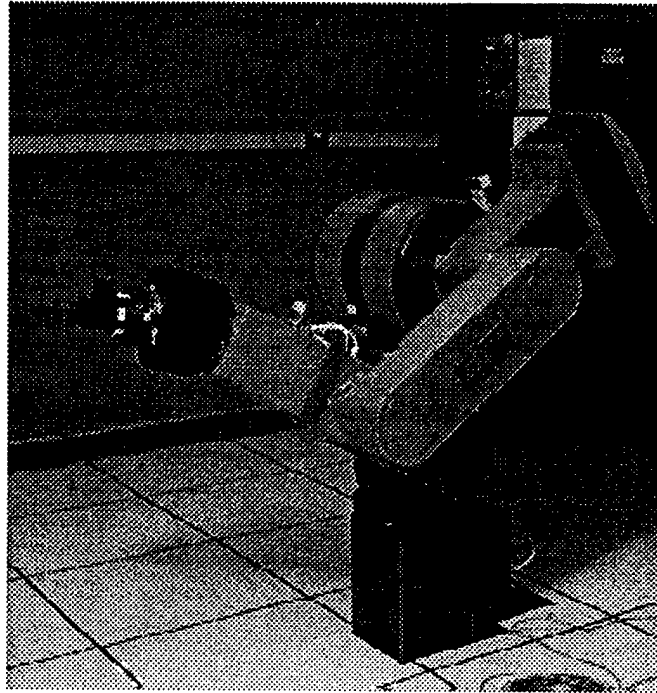
Figure 3. MERLIN Slave Manipulator
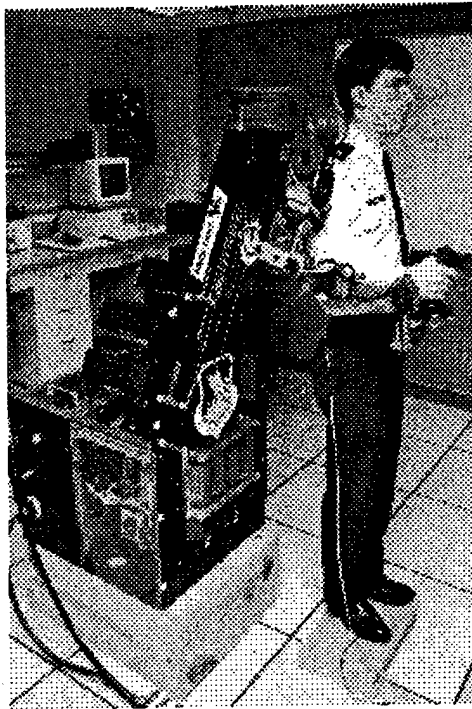
Figure 4. FREFLEX Force-Reflecting Exoskeleton

2.2.1 MERLIN General Description

The MERLIN system includes a six degrees of freedom arm (American Robot Corporation, 1985); controller (American Robot Corporation, 1995b); peripherals which include the teach pendant (American Robot Corporation, 1995a), CRT, and operator's panel; and AR-BASIC (American Robot Corporation, 1996) software.

The left-handed MERLIN 6500 robot arm located in the HSF Laboratory is balanced for a payload of 50 lbs (222.4 N) (American Robot Corporation, 1985). The six joints of the MERLIN are driven by stepper motors. Each time the motors receive a current pulse from the motor drivers, the motors step by 1/25,000 of a motor revolution (American Robot Corporation, 1985). Encoders are mounted on the back of each motor to determine the position of each motor shaft. These encoders read 4000 ticks per revolution and the controller counts the number of motor revolutions to calculate precise joint positions (American Robot Corporation, 1985).

The adjustable mechanical stops; the electromagnetic brakes which prevent the arm from moving when the robot power is off; and the transmissions for the waist, shoulder, and elbow joints are located in the body of the robot. Electrical stops are located in the body for the waist and shoulder joints, while the limit switch for the elbow is located in the upper arm. The transmissions for the wrist axes consist of three concentric tubes that pass through the forearm and transmit torque from the motors to the wrist joints. The range of motion in degrees for joints one through six respectively are 294, 292, 292, ± continuous, ± 90, and ± continuous (American Robot Corporation, 1985).

7

The gear ratios for joints one through six are 48:1, 48:1, 48:1, 24:1, 20:1, and 24:1 respectively (American Robot Corporation, 1985).

The MERLIN controller (American Robot Corporation, 1995b) consists of two central processing units (CPU), the master CPU and the servo CPU. The master CPU is an MSDOS based personal computer (PC) which runs a program that contains the AR-Basic interpreter as well as the motion libraries, ARBASIC.EXE. The servo CPU is actually a single board processor with RAM and VGA that executes SERVO.COM which is responsible for the control of the motors. Both of these CPUs are located in the PC within the American Robot Control Cabinet. The High Speed Host Interface (HSHI) (American Robot Corporation, 1997) is implemented to allow the user to communicate directly with the servo CPU via an RS-232 interface.

The three MERLIN peripherals are the teach pendant, CRT, and the operator's panel. The teach pendant (American Robot Corporation, 1995a), which is connected to the front of the control cabinet, is used for manual control of the MERLIN. The functions of the operator's panel include emergency stop, electromagnetic brake override, robot power, and controller power. The main purpose of the CRT is to display the AR-Basic window. AR-Basic (American Robot Corporation, 1996) is a version of the programming language BASIC developed by American Robot Corporation to create programs for controlling the MERLIN.

2.2.2 FREFLEX General Description

This unique exoskeleton anthropomorphically maps the seven degrees of freedom of the human right arm and provides a natural means to teleoperate a remote manipulator with force-reflection. The human is not attached to the exoskeleton which allows the user to feel less constrained. A forearm push plate with force sensor can be enabled to maintain a constant force on the user's arm to better follow the motions of the human.

The FREFLEX (Odetics, Inc., 1992) is driven by seven brushless p.m. servomotors which are controlled by BDS4 series brushless motor controllers (Industrial Drives, A Kollmorgen Division, 1993). These motors can provide high continuous torque and low armature friction and inertia. Bayside gearheads with reductions ranging from 5:1 to 15:1 are mounted on each motor (Odetics, Inc., 1992). The motors are mounted on an external base minimizing the size, mass, and inertial properties of the FREFLEX exoskeleton.

The seven FREFLEX joints are actuated via these seven base-mounted motors through complex cable-drive systems. The exoskeleton's transmission consists of 19 shafts, 102 pulleys, 92 bearings, and a gear set at the elbow (Odetics, Inc., 1992). The cables, acting as agonist and antagonistic tendons, are routed from the motors to the joints of the exoskeleton via the pulleys mounted at the base. Therefore, the joint angles and joint torques are coupled functions of the motor angles and torques. Huang (1993) used a constant global coupling matrix A to describe the kinematic coupling: $\Delta\Theta_{joint} = A\Delta\Theta_{motor}$. By the principle of virtual work, the same matrix is used in the torque coupling: $\tau_{motor} = A^T \tau_{joint}$. Huang (1993) presented a method to experimentally determine the matrix A using redundant measurements and a least squares fit. Ideally, A is a lower

triangular matrix because motor i torque should only depend on outboard links. Huang experimentally determined A to be

$$A = \begin{bmatrix} -0.9617 & 0.0548 & -0.0169 & 0.0194 & 0.0065 & -0.0024 & -0.0255 \\ 0.9632 & 0.9711 & 0.0039 & 0.0001 & 0.0045 & -0.0029 & -0.0046 \\ -0.0246 & -0.4294 & 0.9672 & 0.0257 & -0.0231 & 0.0177 & -0.0305 \\ 0.0292 & 0.1976 & 0.7246 & -0.7250 & -0.0278 & 0.0310 & -0.0408 \\ 0.0075 & 0.4336 & 0.5890 & 0.3732 & -1.1949 & -0.0052 & -0.0032 \\ -0.0052 & 0.3139 & -0.2185 & 0.2192 & 1.2706 & -1.5005 & -0.0062 \\ 0.0539 & 0.4949 & -0.3120 & 0.3114 & 0.0092 & 1.4494 & -1.4256 \end{bmatrix}. \tag{1}$$

The seven joint angles of the FREFLEX are measured by external potentiometers that are designed into the joint such that the wiper is mounted on the support link and the film is mounted on the driven link. According to Odetics (1992), this arrangement provides better packaging and positive joint angle measurements and requires fewer calibrations. However, operation in the HSF Lab indicates that these pots possess the greatest liability for dependability on the FREFLEX. These pots are unique and have failed in the past. The pot on joint three was not working as designed. A wire has been temporarily added from the brush to the red wire leaving the pot to bypass a discontinuity in the original red wire exiting the pot. To calibrate the pots see Appendix A.

The FREFLEX links' mass m_i and center of mass (CG_i, vector from origin of frame i to CG_i that is expressed in i coordinates) are critical for the FREFLEX gravity compensation algorithm. Odetics (1992) gave conservative values for these parameters, used for worst-case motor sizing. Huang (1993) did not report these parameters, but they appear in the original FREFLEX code and are given in Table 1. Unfortunately, there is no explanation as to how these values were obtained. The current gravity compensation works on the FREFLEX hardware, but it needs improvement. The Table 1 values should be recalibrated: Starting with the seventh link, m_7 and CG_7 should be tuned until each link supports itself against gravity. This process should be repeated for links six through one. For simulation purposes and also future hardware implementation, the FREFLEX code CG_i values (derived for Paul Denavit-Hartenberg parameters) were transformed to equivalent values with the Craig Denavit-Hartenberg (DH) convention; these are reported in Table 2.

Table 1. FREFLEX Mass Parameters, Paul

Link	m_i lbsm (kg)	CG_{ix} inches (cm)	CG_{iy} inches (cm)	CG_{iz} inches (cm)
1	0.02163	1.15	-7.28	-4.40
	(0.00981)	(2.92)	(-18.49)	(-11.18)
2	0.01684	0.00	1.58	7.55
	(0.00764)	(0.00)	(4.01)	(19.18)
3	0.00958	-1.57	-1.36	-0.09
	(0.00435)	(-3.99)	(-3.45)	(-0.23)
4	0.01036	0.17	0.10	2.62
	(0.00470)	(0.43)	(0.25)	(6.65)
5	0.00389	0.00	-3.69	-3.90
	(0.00176)	(0.00)	(-9.37)	(-9.91)
6	0.00285	0.00	-4.00	-3.50
	(0.00129)	(0.00)	(-10.16)	(-8.89)
7	0.00437	-0.58	0.00	-2.60
	(0.00198)	(-1.47)	(0.00)	(-6.60)

Table 2. FREFLEX Mass Parameters, Craig

Link	m_i lbsm (kg)	CG_{ix} inches (cm)	CG_{iy} inches (cm)	CG_{iz} inches (cm)
1	0.02163	1.15	4.40	-7.28
	(0.00981)	(2.92)	(11.18)	(-18.49)
2	0.01684	0.00	5.75	2.41
	(0.00764)	(0.00)	(14.61)	(6.12)
3	0.00958	0.40	0.76	-1.13
	(0.00435)	(1.02)	(1.93)	(-2.87)
4	0.01036	-1.80	2.50	0.80
	(0.00470)	(-4.57)	(6.35)	(2.03)
5	0.00389	0.00	2.40	-4.80
	(0.00176)	(0.00)	(6.10)	(-12.19)
6	0.00285	0.00	3.50	-4.00
	(0.00129)	(0.00)	(8.89)	(-10.16)
7	0.00437	1.10	0.00	-2.60
	(0.00198)	(2.79)	(0.00)	(-6.60)

The FREFLEX VME chassis contains four VME based processors and I/0 boards mounted in a 21 slot chassis. The VMIC 4100 board outputs voltages to the FREFLEX motor controllers. The VMIC 2510B board provides discrete input and output channels

for the exoskeleton operator interface. The Ironics IV-3230 board labeled *Crusher* is used for force-reflection processing while the Ironics IV-3230 board *Control* is the master real-time processing unit. The chassis also contains a JR^3 board that processes information from the JR^3 force and torque sensor mounted at the wrist of the FREFLEX, a Data Translation DT1401 card that reads the pots, and a Bit 3 card that links the ethernet and the Sun SPARCstation.

2.2.3 FREFLEX and MERLIN Kinematics

The telerobotic control architecture presented in this report requires kinematics transformations which relate Cartesian and joint variables within the master and slave devices. Specifically, this section presents the Denavit-Hartenberg parameters, forward kinematics transformation, and *Jacobian* matrices for the FREFLEX master and MERLIN slave. Huang (1993) presented these equations derived for minimal on-line computation. The equations in this section, used in the simulation and hardware implementation, do not attempt symbolical or numerical computation optimization. Numerical recursion is used. Huang's equations were implemented on the FREFLEX hardware and our hardware implementation makes use of that existing code insofar as possible. Huang's equations were never implemented on the MERLIN and so the equations from this report are used. Computational efficiency can be improved over the equations presented in this report by using symbolic computer algebra to derive the equations or a numerical approach similar to Huang's.

The Denavit-Hartenberg parameters provide a standard manner to describe the joint and link geometric relationships in a serial manipulator. Unfortunately, two possible DH standards have arisen, the Paul (Paul, 1981) convention and the Craig (Craig, 1989) convention. Craig convention is used in the current report and in the Odetics (1992) report, while Huang (1993) used Paul convention. The convention selection is based on personal preference. Figure 5 shows the MERLIN and Figure 6 the FREFLEX kinematic diagrams, from which the DH parameters (Craig, 1989) of Tables 3 and 4 are derived. All angular units are degrees. If the θ_i angular offset of rows 3 and 6 are included for the MERLIN and FREFLEX, respectively, Figures 5 and 6 show the zero-joint-angle configurations. Nominal MERLIN and measured FREFLEX joint angle limits are also given.

Nominal values for the MERLIN lengths are: $a_2 = 17.375$ inches (44.1325 cm), $d_2 = 11.9$ inches (30.226 cm), $d_4 = 17.25$ inches (43.815 cm), and nominal values for the FREFLEX lengths are: $a_3 = 1.969$ inches (5.00126 cm), $a_4 = -1.969$ inches (-5.00126 cm), $d_3 = 14.64$ inches (37.1856 cm), $d_4 = 0.625$ (1.5875 cm), $d_5 = 11.77$ inches (29.8958 cm).

Table 3. MERLIN DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i	Limits
1	0	0	0	θ_1	±147
2	-90	0	d_2	θ_2	+56,-230
3	0	a_2	0	$\theta_3 - 90$	+56,-230
4	-90	0	d_4	θ_4	±360
5	90	0	0	θ_5	±90
6	-90	0	0	θ_6	±360

Table 4. FREFLEX DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i	Limits
1	0	0	0	θ_1	18,-28
2	90	0	0	θ_2	+130,-52
3	-120	0	d_3	θ_3	±90
4	120	a_3	d_4	θ_4	-3,-166
5	-70	a_4	d_5	θ_5	±90
6	70	0	0	$\theta_6 + 90$	+128,+51
7	90	0	0	θ_7	+57,-52
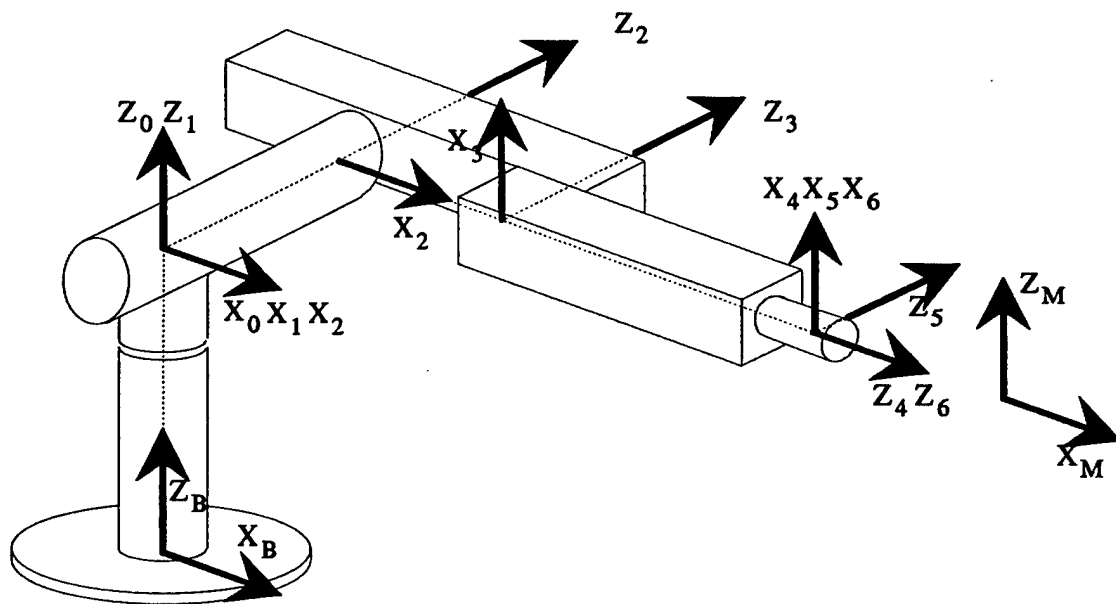
Figure 5. MERLIN Kinematic Diagram
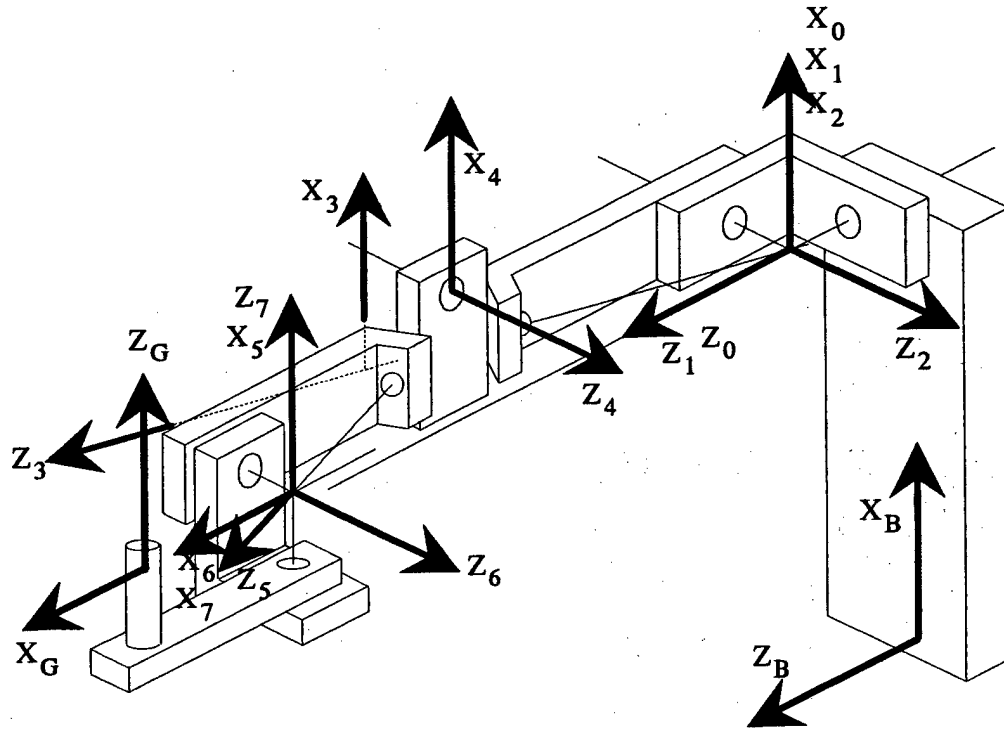
Figure 6. FREFLEX Kinematic Diagram

The forward kinematics transformation gives the position and orientation of the moving frame of interest n with respect to the kinematic base frame 0 (Craig, 1989):

$$^0_nT = \begin{bmatrix} \begin{bmatrix} ^0_nR \end{bmatrix} & \begin{Bmatrix} ^0P_n \end{Bmatrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \qquad ^0X_n = \{x \quad y \quad z \quad \gamma \quad \beta \quad \alpha\}^T. \tag{2}$$

The pose can be represented in two ways. The first method of pose representation is by ^0_nT, which is the 4x4 homogeneous transformation matrix with the 3x3 orientation matrix ^0_nR and the 3x1 position vector ^0P_n. The second pose representation method is the vector, ^0X_n, whose first 3 components are ^0P_n and second 3 are orientation numbers extracted from ^0_nR, e.g. Z-Y-X Euler convention, see Equation 11. Given one row in a DH parameter table, the homogeneous transformation matrix relating the pose of neighboring frames in a serial chain is

$$^{i-1}_iT = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

where $c\theta_i = \cos(\theta_i)$, $s\theta_i = \sin(\theta_i)$, etc. The forward kinematics transformation for active joints is

$$
{}^{0}_{n}T = \prod_{i=1}^{n} {}^{i-1}_{i}T = {}^{0}_{1}T\,{}^{1}_{2}T \cdots {}^{n-1}_{n}T .
\tag{4}
$$

Note the MERLIN and FREFLEX each have Base, 0, and Wrist Frames which must be distinguished, see Figure 1. The overall forward kinematics for the MERLIN and FREFLEX are given below on the left and right, respectively:

$$
{}^{Wo}_{M}T = {}^{Wo}_{B}T\,{}^{B}_{0}T\,{}^{0}_{W}T\,{}^{W}_{M}T \qquad \text{and} \qquad {}^{Wo}_{G}T = {}^{Wo}_{B}T\,{}^{B}_{0}T\,{}^{0}_{W}T\,{}^{W}_{G}T .
\tag{5}
$$

where M, B, 0, W, and G stand for the MRF, Base, 0, Wrist, and Grip Frames. The World Frame, Wo, is common.

The Jacobian matrix ${}^{k}J$ for a serial chain maps joint rates $\dot{\Theta} = \left\{ \dot{\theta}_1 \quad \dot{\theta}_2 \quad \cdots \quad \dot{\theta}_n \right\}^{T}$ into Cartesian rates ${}^{k}\dot{X} = \left\{ \dot{x} \quad \dot{y} \quad \dot{z} \quad \omega_x \quad \omega_y \quad \omega_z \right\}^{T}$ of the frame of interest with respect to the base, expressed in any frame k: ${}^{k}\dot{X} = {}^{k}J\dot{\Theta}$. The ith column of ${}^{k}J$ is the Cartesian velocity of the point of interest due to joint rate i alone with $\dot{\theta}_i$ factored out. This fact leads to the following formula for the ith column of ${}^{k}J$, where ${}^{i}z_i = \left\{ 0 \quad 0 \quad 1 \right\}^{T}$:

$$
{}^{k}J_i = \left\{ \begin{matrix} {}^{k}_{i}R\left({}^{i}z_i \times {}^{i}P_n \right) \\ {}^{k}_{i}R\,{}^{i}z_i \end{matrix} \right\} .
\tag{6}
$$

Equation 6 is applied for each moving joint to yield the 6x6 MERLIN and 6x7 FREFLEX Jacobian matrices, each relating the motion of the respective Wrist Frame with respect to the Base Frame, expressed in k. Note that k can be different for MERLIN and FREFLEX and is chosen as the respective 0 Frames in this report.

2.2.4 Chimera 3.2 General Description

This section of the report will summarize the necessary material concerning the Chimera 3.2 real-time operating system needed to operate the MERLIN manipulator via the FREFLEX exoskeleton. Note that an understanding of UNIX commands is essential since Chimera 3.2 is accessed from a Sun SPARCstation with a UNIX operating system.

The base directory for controlling the MERLIN manipulator via the FREFLEX master is named freflex_merlin and it is located at /usr/chimera/chim_3.2/freflex_merlin. Chimera 3.2 defines a subdirectory hierarchy that all files under the base directory must follow (Ingimarson et al., 1995). Figure 7 shows this hierarchy under freflex_merlin.

14
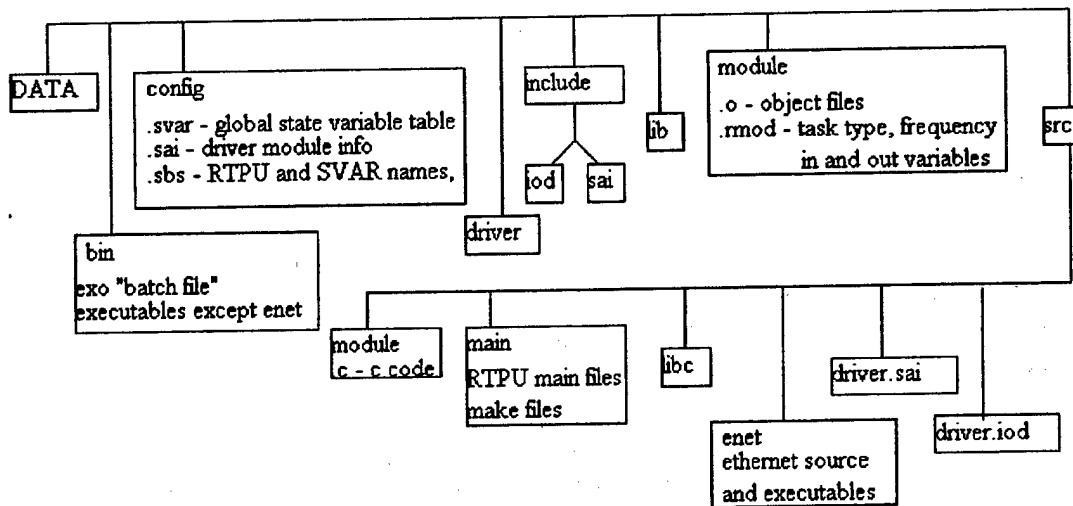
Figure 7. Freflex_Merlin Directory Hierarchy

There are two file editors available on the UNIX system, these are *VI* and *Nedit*. To compile a single module use the makefile in the subdirectory in which the file is located, adding the filename if not already on the list of object files to be created within the makefile. Type "make *filename*.o" to compile a single module. If ".o" is not applied to the end of the filename an error will result. To compile and link all modules for RTPU Crusher type "make -f freflex_merlin_rtpu.mak all". Change the directory to /usr/chimera/chim_3.2/freflex_merlin/src/main, remove freflex_merlin.o by typing "rm freflex_merlin.o", and type "make all" when compiling and linking all modules for RTPU Control. A Chimera 3.2 library with helpful functions, including a matrix math library which was used extensively in the freflex_merlin development, is available (Ingimarson et al., 1995).

Each periodic task is composed of several components as required by Chimera 3.2. These components are the C subroutines: *init, on, cycle, off, kill, clear, set,* and *get* (Ingimarson et al., 1995). Good examples of the subsystem configuration are the existing ".c" files in the /usr/chimera/chim_3.2/freflex_merlin/src/module directory. When operating Chimera 3.2 there is a subsystem interactive command interpreter that is useful. It accepts the commands: *display, kill, module, off, on, quit, set, spawn,* and *status* (Ingimarson, et al., 1995). Details of these commands can be found in the Chimera 3.2 manual or by typing "help" when operating Chimera 3.2.

Chimera 3.2 uses a global state variable table (SVAR), which is stored in a configuration file, to communicate between different modules when operating in a reconfigurable multiprocessor environment (Ingimarson et al., 1995). Each task creates and modifies a local copy of the SVAR and periodically returns these global variables, which are used by different tasks, back to the SVAR.

2.2.5 JR³ Universal Force-Moment Sensor System General Description

The JR³ sensors are necessary for wrench reflection. The sensor located on the wrist of the MERLIN measures the forces applied by the robot end effector. The sensor located on the FREFLEX was used in past taskboard experiments but is not used in the current study. For this reason, the description of these devices will be concentrated on the MERLIN mounted sensor.

The MERLIN mounted JR³ Universal Force-Moment Sensor System, serial number 0310 (JR³, Inc., 1988), consists of one sensor body, one sensor cable that connects the sensor to a data processing electronics enclosure, a JR³ power box, and a serial cable to port #1 on the RTPU Control in the VME chassis.

Foil strain gauges in the sensor body detect forces and moments and send millivolt signals to the electronic processing enclosure through the sensor cable (JR³, Inc., 1988). These signals are amplified and processed within the electronic enclosure before being sent to Control via the serial cable. The electronic enclosure can also receive commands through the serial line from software operating on the RTPU Control.

The force load ratings for the JR³ sensor located at the MERLIN wrist are 25.0 lbs (111.2 N) for the X and Y axes and 50.0 lbs (222.4 N) for the Z axis (JR³, Inc., 1988). The torque load ratings are 100.0 in•lbs (11.3 N•m) about each Cartesian axis (JR³, Inc., 1988). Violating these ratings could damage the sensor and hamper proper force reflection. Also note that the JR³ sensor located at the MERLIN wrist has a left-handed coordinate system, which is contrary to the JR³ manual description.

The JR³ sensor located on the FREFLEX is newer and slightly different than the sensor mounted on the wrist of the MERLIN. The FREFLEX mounted sensor does not have its own electronics enclosures. Instead it is connected to its own VME board by a line with a modular plug on each end. In the future this sensor could be useful to compare the force applied by the teleoperator to the force exerted by the slave manipulator.

2.2.6 Taskboard General Description

The standardized peg-in-hole taskboard which is shown in Figure 8 was designed and built by the Naval Ocean Systems Center (Spain, 1989).

16
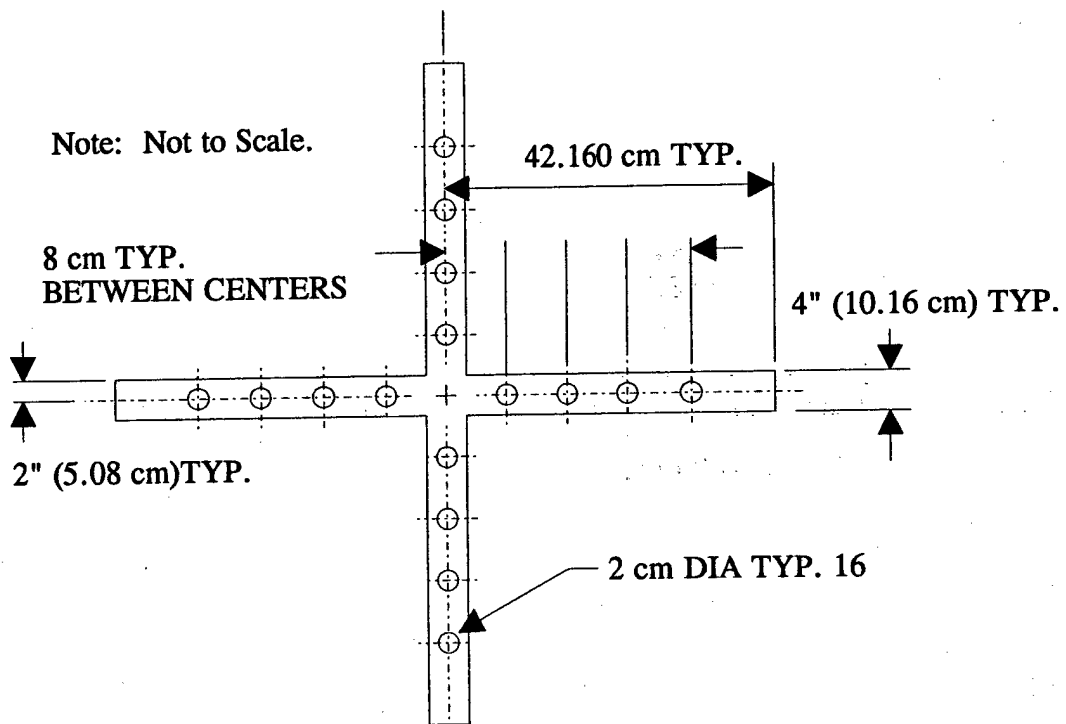
Figure 8. Peg-in-Hole Taskboard

The taskboard is complete with switches in the bottom of each of the 16 holes in the board. When a peg is inserted into the hole, the switch in that particular hole is activated. This allows precise time measurement of the movement from the start hole to the finish hole of a task.

The peg and hole sizes can be changed to increase or decrease tolerances. The distance between the start and finish holes and the respective tolerance allows for varying degrees of task difficulty to be tested.

Since the taskboard can be operated either via teleoperation or directly by a human, it can be used to define many teleoperation performance measures. Comparisons can be made between teleoperation and direct human operation, different operator interfaces, or different teleoperation control algorithms.

CHAPTER 3 CONTROL ARCHITECTURE

This chapter presents the general real-time telerobotic control architecture for implementation in the HSF Lab. It is adapted from a unique hardware system at NASA Langley (Williams et al., 1997). The sensor-rich system is rate based but allows input from various control modes simultaneously: pose, rate, vision, force; others may be added. The architecture provides shared telerobotic control, defined as concomitant human, automated, and sensor-based inputs. All modes can operate on all Cartesian axes simultaneously and this can lead to conflicts of which the operator must be aware. During tasks one or more control modes may be activated during task steps by entering nonzero gain matrices. Joint control, pose control, force control, and rate control are implemented for a single slave manipulator and force-reflecting master in the current report. This can be expanded in the future as needed for additional sensory feedback modes such as machine vision and laser proximity, dual-slave-arm operations, or kinematically-redundant slave manipulators. Huang's (1993) control mode for the slave manipulator allowed only inverse pose control and was based on the complex, multiple solution inverse pose kinematics results. The current rate-based method does not use these equations, but instead requires the slave manipulator Jacobian matrix. Rate control has several benefits such as linear equations, unique solution, and inputs from multiple control modes are linearly summed. However, it is not as widely applied as inverse pose control. Both inverse rate and inverse pose suffer from the same manipulator kinematic singularities.

3.1 Control Diagram

The real-time, sensor-rich, rate-based, shared telerobotic control architecture is shown in Figure 9 for a single slave manipulator. The following subsections present the control modes and algorithms.
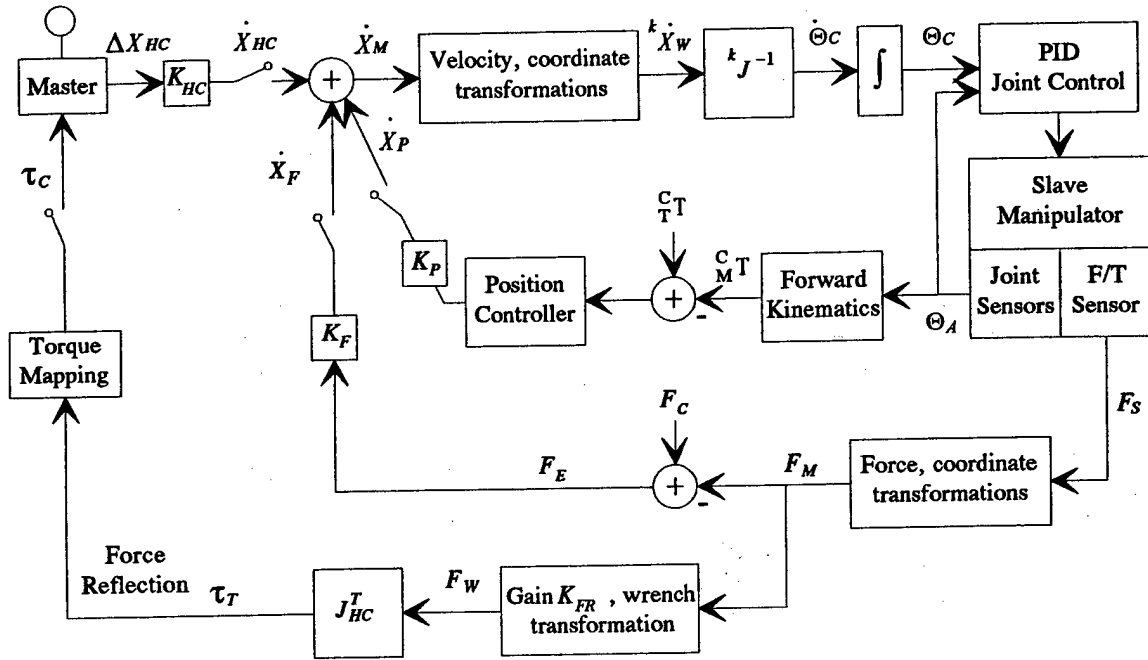
Figure 9. Telerobotic Control Architecture

3.1.1 Resolved-Rate Control

The resolved-rate control algorithm is used for motion control from all input sources: master, pose, and force controllers. The algorithm implemented is based on Whitney's method (Whitney, 1969). This section assumes a static Base Frame and CRF; the method can be extended to handle moving Base Frames and CRFs for dynamic tasks. The time-varying manipulator Jacobian matrix maps joint rates to Cartesian rates of the Wrist Frame: $^k\dot{X}_W = {}^kJ\dot{\Theta}$. The Cartesian rates $^k\dot{X}_W = {}^k\{\underline{v}_W \quad \underline{\omega}_W\}^T$ express the translational and rotational velocities of the Wrist Frame with respect to the Base Frame, expressed in the coordinates of any frame k. Common choices are k = Wrist Frame, 0 Frame, or Base Frame; simplest symbolic terms for the Jacobian matrix result when k is the frame midway between the Base Frame and Wrist Frame, often the Elbow Frame. The equation $^k\dot{X}_W = {}^kJ\dot{\Theta}$ must be inverted (or more efficiently, solved for $\dot{\Theta}$ by Gaussian elimination) at each control step. First, however, the input MRF Cartesian rates $^j\dot{X}_M$ (the sum of all control inputs for the MRF, expressed in any frame j) must be converted to the resolved rate input $^k\dot{X}_W$ (the equivalent Cartesian velocities of the Wrist Frame to produce $\dot{X}_M$). This rigid-body velocity transformation and coordinate transformation is given in Equation 7 (Craig, 1989):

$$^k\dot{X}_W = {}^k\begin{Bmatrix} \underline{v}_W \\ \underline{\omega}_W \end{Bmatrix} = \begin{bmatrix} {}^k_jR & {}^k_WR \, {}^WP_{MRF} \times {}^k_jR \\ 0 & {}^k_jR \end{bmatrix} {}^j\begin{Bmatrix} \underline{v}_{MRF} \\ \underline{\omega}_{MRF} \end{Bmatrix}. \tag{7}$$

$\dot{X}_M$ always gives the six degrees of freedom velocity of the MRF with respect to the Base Frame, but can be expressed in any coordinates j such as the CRF, Base Frame, 0 Frame, or World Frame. Now the rate equation is inverted to calculate the instantaneous joint rates necessary to obtain the commanded $^k\dot{X}_W$:

$$\dot{\Theta}_C = {}^kJ^{-1}\,{}^k\dot{X}_W. \tag{8}$$

The commanded joint rates are numerically integrated to obtain the commanded joint angles Θ_C. These angles are ordered to the manipulator and achieved using linear independent *PID* control laws. Joint encoder feedback Θ_A is used to form the errors for servo control.

This algorithm is sensitive to kinematic singularities, where the manipulator loses freedom to move in one or more Cartesian directions. In the neighborhood of singularities, extremely high joint rates are theoretically required to satisfy a finite Cartesian command. To deal with this problem, the determinant of the Jacobian matrix ^kJ must be monitored. When the determinant approaches zero, the matrix inverse (or Gaussian elimination) in Equation 8 is replaced by a matrix pseudoinverse based on Singular Value Decomposition (SVD). Near singularities, the exact Cartesian command $^k\dot{X}_W$ cannot be satisfied, but the SVD with the damped least-squares formulation will yield bounded joint rates which will move the manipulator through the singular neighborhood until Equation 8 is valid again (Maciejewski & Klein, 1989).

For teleoperation, the displacement ΔX_{HC} of the operator's hand with the master device is interpreted to be the rate $\dot{X}_{HC}$ after applying matrix gain K_{HC}, as discussed in the FRHC Cartesian Input Command section discussed later.

3.1.2 Pose Control

Resolved-rate control may be used to command manipulator poses by closing a position loop around the rate system. The difference between the commanded, ^C_TT (frame T stands for Target), and current, ^C_MT, manipulator poses must be calculated. The target pose may be commanded by the operator through teleoperation or keyboard input, an automated path planner, or some sensor-based algorithm (e.g. machine vision). The current pose is found from the forward kinematics transformation of joint encoder feedback Θ_A and other known homogeneous transformation matrices:

$$^C_MT = {}^C_{WO}T\,{}^{WO}_BT\,{}^B_WT(\Theta_A)^W_MT. \tag{9}$$

The translational error vector is found by algebraic subtraction of the position vectors: $^CP_{T-M} = {}^CP_T - {}^CP_M$. However, because the orientation cannot be represented by vectors the angular velocity error must be calculated using a rotation matrix "difference":

$$\begin{smallmatrix}M\\T\end{smallmatrix}R = \begin{smallmatrix}C\\M\end{smallmatrix}R^{-1}\begin{smallmatrix}C\\T\end{smallmatrix}R = \begin{smallmatrix}C\\M\end{smallmatrix}R^T\begin{smallmatrix}C\\T\end{smallmatrix}R .\tag{10}$$

Three orientation numbers (e.g. Euler Z-Y-X α,β,γ, [Craig, 1989]) are extracted from the difference rotation matrix $\begin{smallmatrix}M\\T\end{smallmatrix}R$, as given in Equation 11:

$$R = [r_{ij}] = \begin{bmatrix} c\alpha c\beta & -s\alpha c\gamma + c\alpha s\beta s\gamma & s\alpha s\gamma + c\alpha s\beta c\gamma \\ s\alpha c\beta & c\alpha c\gamma + s\alpha s\beta s\gamma & -c\alpha s\gamma + s\alpha s\beta c\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad \begin{array}{l} \beta = a\tan 2\left(-r_{31}, \pm\sqrt{r_{11}^2 + r_{21}^2}\right) \\[6pt] \alpha = a\tan 2(r_{21}/c\beta, r_{11}/c\beta) \\[6pt] \gamma = a\tan 2(r_{32}/c\beta, r_{33}/c\beta) \end{array} \tag{11}$$

where $c\beta = \cos(\beta)$, $s\beta = \sin(\beta)$, $c\alpha = \cos(\alpha)$, $s\alpha = \sin(\alpha)$, $c\gamma = \cos(\gamma)$, and $s\gamma = \sin(\gamma)$. The α,β,γ solution in Equation 10 has two results represented by the $\pm$ in the β solution. The solution is subject to a $\beta = \pm 90°$ singularity (Craig [1989] presents an alternative solution for the singular case). Taking these three numbers α,β,γ as both the Euler angles and respective rates ($\dot\gamma = \gamma$, $\dot\beta = \beta$, and $\dot\alpha = \alpha$), the commanded angular velocity error vector is calculated using the appropriate rotational kinematic differential equations in Equation 12 (Kane, Likins, & Levinson, 1983):

$$\begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -s\beta \\ 0 & c\gamma & c\beta s\gamma \\ 0 & -s\gamma & c\beta c\gamma \end{bmatrix} \begin{Bmatrix} \dot\gamma \\ \dot\beta \\ \dot\alpha \end{Bmatrix}.\tag{12}$$

The position and orientation error vector is converted to a rate $\dot{X}_P$, added into the summing junction in Figure 10, after applying the vector gain K_P that has translational units $1/s$ and unitless rotational components.

3.1.3 Force Control

An active force controller has been implemented in the resolved-rate scheme to command forces to the environment with the manipulator. This active force controller is basically a general impedance controller (Hogan, 1985) with only the damping term. A six degrees of freedom force and torque sensor (with frame S) mounted after the last joint reads the contact wrench $F_S = \{f_S \quad m_S\}^T$. The weight and gravity moment of the end effector mounted outboard of the force and torque sensor (transformed to the S Frame) must be subtracted from the sensor reading. This modified sensor reading in the S Frame must be transformed by rigid body transformations and coordinate rotations (Craig, 1989) to the equivalent MRF wrench:

$$F_M = \begin{Bmatrix} f_M \\ m_M \end{Bmatrix} = \begin{bmatrix} \begin{smallmatrix}M\\S\end{smallmatrix}R & 0 \\ {}^M P_S \times \begin{smallmatrix}M\\S\end{smallmatrix}R & \begin{smallmatrix}M\\S\end{smallmatrix}R \end{bmatrix} \begin{Bmatrix} f_S \\ m_S \end{Bmatrix}.\tag{13}$$

A wrench error vector $F_E = F_C - F_M$ is formed from the difference of the sensed and commanded wrenches in the MRF. Since both force and moment are vector quantities, algebraic subtraction applies. The wrench error is converted to a rate $\dot{X}_F = K_F F_E$ which is sent to the summing junction in Figure 10. This rate drives the manipulator motion so the desired force is achieved continuously. The diagonal gain matrix K_F has units m/Ns and radians/Nms for translational and rotational terms, respectively. If zero wrench is commanded and the manipulator contacts the environment, the motion will automatically align the manipulator end-effector for minimal Cartesian contact wrench and misalignments. This is called *force and moment accommodation* (FMA) (Williams et al., 1997; Willshire et al., 1992).

If a FRHC is used, the sensed MRF wrench can also be applied by the FRHC so that the operator's hand feels the task wrench exerted by the manipulator. The required transformation is (Craig, 1989)

$$\tau_T = J_{HC}^T F_W . \tag{14}$$

where τ_T is the vector of FRHC joint forces and torques required to feel the task wrench and J_{HC} is the FRHC Jacobian matrix. The task wrench F_M is scaled by matrix gain K_{FR} and sent as the FRHC grip wrench. If the FRHC Jacobian is derived for the Wrist Frame relative to the Base Frame, a rigid body wrench transformation (similar to Equation 13) is required to transform this scaled task wrench from the Grip Frame to the Wrist Frame, obtaining F_W for use in Equation 14. The joint torques are achieved by torque mapping, sending τ_C to the FRHC.

3.1.4 Simultaneous Control

In the control architecture of Figure 9, all input sources (i.e. master, pose, and force) can be enabled simultaneously for all Cartesian axes. In most other experimental telerobotic systems that the author is acquainted with, only one input source is enabled at any one time and changing between sources requires artificial software or hardware switches. Often different input sources will result in competing goals (e.g. different poses commanded by the operator and automated path planner). Therefore, software switches are included (set by script file keyboard input) to enable or disable each input source during the execution of tasks. Also, zero values in the vector gains K_{HC}, K_P, and K_F can be used to disable some or all Cartesian axes from the input sources. The NTRFC is an example of simultaneous inputs from two control modes which is complimentary, not competing. A limitation of the proposed control architecture is that the gains are tuned heuristically. Gain scheduling is allowed but there is no theoretical basis for computing the gains of complex systems. As the manual gain selection is necessarily conservative to achieve stability, it is likely that suboptimal performance is obtained. Methods for gain selection for simple models are demonstrated in Chapter 4.

3.1.5 Shared Control

The proposed control architecture allows shared control, which is control by a human operator (teleoperation), autonomous sensor-based control (robotic), or a combination of both (telerobotic). In this system the human controls the system via the master or through keyboard inputs. The master input is integrated seamlessly. For instance, if it appears the automated system will drive the end effector into an obstacle the operator can modify the trajectory in real-time by using the master. After the danger is past and the master input is zero the original target pose is still reached by the manipulator.

3.1.6 Joint Control

Joint control can be implemented for the slave manipulator by commanding Θ_c directly. This mode is useful for gross changes in the slave configuration, but other Cartesian modes must be disabled during joint control. Joints may be moved one at a time in either absolute joint angle displacements or rates.

3.2 Force-Reflecting Master Control Diagram

Figure 10 shows the control flow for the implementation of a Cartesian FRHC commanding inputs and reflecting wrenches with a telerobotic system in Cartesian space. There is some overlap between Figures 9 and 10; Figure 10 shows more detail. Figure 10 assumes Cartesian rate inputs; the difference for Cartesian pose inputs is minor (discussed below). The following subsections present the algorithms for Figure 10. For more detail, see (Williams, 1997).

Scale | τRD | J_{HC}^T | Grip to W Wrench Transform | $^WF_{RD}$ | $^GF_{RD}$ | ^GF_R | ^GF_D | K_D

τRDs

$+$ | τG | FRHC Grav Comp | K_R | $\Delta(\Delta X_{HC})$

τRDG | ΔX_{HC} | Change in ΔX_{HC}

$+$ | τ | Saturate Gear Ratios Torque Curves | τC | FRHC Including Operator | Θ | FRHC Kinematics | ^0_GT | TMAT Difference | ΔX_{HC} | K_{HC}

τT | $^0_{G_0}T$ | $\dot X_{HC}$

Define Initial Pose

J_{HC}^T | F_W | Grip to W Wrench Transform | F_G | K_{FR} | F_M | S to MRF Wrench Transform | F_T | EE, Load Grav Comp | F_S | Robot and F/T Sensor

Figure 10. Cartesian FRHC Control Diagram

3.2.1 FRHC Cartesian Input Commands

Let us start with the block "FRHC Including Operator" in Figure 10. The user must first define the desired FRHC reference pose via a switch. This pose (calculated from forward kinematics $^0_{G_0}T = ^0_1T^1_2T \cdots ^6_7T^7_{G_0}T$ when the switch is hit) can be any convenient pose in the FRHC workspace which represents zero Cartesian input to the manipulator. The operator may redefine this pose at any time. This feature is intended to decouple the Cartesian input from the FRHC base frame and allow generality for commands.

During any control cycle when the operator has moved the FRHC grip from the reference pose, the input command to the manipulator is determined as follows. First the FRHC joint sensors are read and FRHC forward kinematics calculates the current grip frame pose relative to the FRHC base frame: $^0_GT = ^0_1T^1_2T \cdots ^6_7T^7_GT$. Now a difference homogeneous transformation matrix is calculated to represent the input (the inverse is given by Craig [1989]):

$$^{G_0}_GT = ^0_{G_0}T^{-1}\,^0_GT \,. \tag{15}$$

At this point, the Cartesian pose and rate input cases differ.

The difference matrix $^{G_0}_GT$ is interpreted as the commanded slave pose $^{M_0}_TT$ (where M_0 is the reference MRF, which can also be redefined using FRHC pose indexing). The frame M_0 is fixed once defined, as opposed to the constantly changing MRF in pose control (Equation 9). The pose input is then $^C_TT = ^C_{M_0}T^{M_0}_TT = ^C_{M_0}T^{G_0}_GT$. This is not pictured in Figure 10.

This procedure is similar to formation of rate signal $\dot{X}_P$ for resolved-rate-based pose control. A set of difference numbers $\Delta X_{HC} = \{x \quad y \quad z \quad \gamma \quad \beta \quad \alpha\}^T$ is extracted from the difference matrix $^{G_0}_G T$. The translational part $\{x \quad y \quad z\}^T$ is the fourth column of $^{G_0}_G T$ excluding row 4, while α, β, γ are the Z-Y-X Euler angles (Craig, 1989) extracted from the difference rotation matrix $^{G_0}_G R$. This is labeled as "*TMAT* Difference" in Figure 10. For Cartesian rate input, the translational terms are the first three terms of ΔX_{HC} scaled by the first three K_{HC} diagonal elements (units s^{-1}). In Figure 10 all gain matrices K_m are order 6x6 and (generally) diagonal matrices of gains. Because a static FRHC orientation must be converted into a Cartesian rotational rate, we again use $\dot{\gamma} = \gamma$, $\dot{\beta} = \beta$, and $\dot{\alpha} = \alpha$ in the rotational rate kinematics transformation Equation 12. Then the angular velocity command is $\{\omega_x \quad \omega_y \quad \omega_z\}^T$, scaled by the second three K_{HC} diagonal elements (unitless). The total teleoperated Cartesian rate command to the *MRF* ($\dot{X}_{HC}$) is formed from these translational and rotational rate terms. Note this process is not shown in Figure 10, but is inside the K_{HC} block.

The lower path in Figure 10 presents reflection of the Cartesian task wrench to the operator. Previously-presented algorithms are sufficient to command a manipulator and feed back task wrenches simultaneously with a FRHC in Cartesian space. However, the next section presents the additional features of Figure 10 to improve operator loading and FRHC stability.

3.2.2 Improved Operator Loading and Stability

One benefit of wrench reflection to the operator is increased feeling of telepresence which enables teleoperation tasks to be completed more easily and with lower contact wrenches. However, one drawback is the potential for increased operator loading, including fatigue from resisting wrenches through the FRHC and supporting a portion of the FRHC weight. In the last section, the end effector and payload weights and moments were subtracted to remove that static loading from the operator (this information is not required to complete tasks). However, to improve telepresence quality, the payload weight may be included if desired.

In the current section, FRHC gravity compensation is presented to further unload the operator's arm. Also, for Cartesian rate inputs, a unique return-to-center method is developed which assists the operator in finding the zero-input FRHC pose when zero inputs are desired in between commanded motions. For both rate and pose inputs, a damping term is also added to improve relative FRHC stability. It is crucial that these operator aids do not mask the task Cartesian contact wrench.

Many FRHCs are mini-articulated robots which must be supported by the operator. FRHC gravity compensation applies configuration-varying joint torques so that the FRHC supports most (theoretically, all) of its static weight. The weight $m_i g$ acts at the center of gravity of the FRHC link i. If a fictitious force $f_{icomp} = m_i g$ is provided equal and opposite of the weight vector, that link will be balanced. The joint torques required to

support this f_{icomp} may be calculated using $\tau_i = J_i^T f_{icomp}$, where J_i is the Jacobian matrix relating the center of mass of link i to the base. Only motors one through i support the weight of link i. By summing all link's τ_i (vectors of increasing dimension 1 through n for links 1 through n) we calculate the joint torques τ_G in Figure 10 required to unload the operator's arm by commanding the FRHC to support its own weight. Huang (1993) presented an alternative gravity compensation algorithm; a simulation demonstrated identical results for the two methods. The nominal mass and mass center parameters were presented in Table 1 and Table 2.

For Cartesian rate input commands, the manipulator will move with a commanded velocity when the FRHC Cartesian pose is different from its reference pose G_0. Therefore, a return-to-center (RTC) force should be provided to assist the operator's hand in finding the zero-input FRHC pose. As a first try, the RTC was calculated at NASA Langley Research Center using Hooke's law with a virtual spring (Williams, 1997): $^G F_R = -K_R \Delta X_{HC}$. The FRHC grip wrench is calculated for each Cartesian axis (three translations, three rotations) independently; the negative sign is to draw the operator's hand back toward the zero pose. However, it was found at NASA (Williams, 1997) that the FRHC workspace that was far from the defined reference pose generated unnecessarily large RTC forces due to the linearly increasing relationship.

Therefore, a novel constant-force return-to-center (CFRTC) approach was developed. Figure 11 shows the CFRTC force as a function of scalar displacement ΔX_{HCi} from the zero reference, for one of the six Cartesian axes. The ΔX_{HCi} represents any one of the six terms in the relative Cartesian pose ΔX_{HC}. The ΔX_{HCi} (i = 1, 2, ..., 6) terms are found from FRHC joint sensors as discussed previously. The magnitudes in Figure 11 are arbitrary and must be determined for specific FRHCs based on performance requirements and FRHC workspace. The CFTRC is symmetric about $\Delta X_{HCi} = 0$; each side displays three distinct (but continuous) regions. The first is the deadband and serves two purposes: a) providing a small region of zero input surrounding the zero pose and b) providing a parabolic virtual wall which the operator must overcome if an input is to be commanded in that particular Cartesian axis. The second, largest, zone is the working range which provides the CFRTC (as opposed to Hooke's law) virtual spring. The third zone provides a stiff virtual spring to alert the operator when the edge of the FRHC workspace is encountered. It was found at NASA that this stiff spring was unnecessary (Williams, 1997). Therefore the flat CFRTC zone was extended to the workspace boundary. In this case, the operator must be aware of the workspace boundaries, but the effective FRHC range is extended.
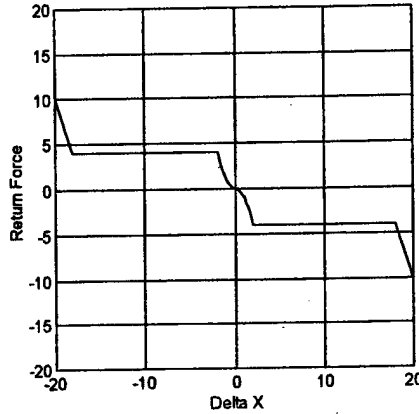
Figure 11. CFRTC for One Cartesian Axis

The ith term for the Cartesian CFRTC wrench $^{G}F_R$ is expressed in Equation 16. Note translational pose terms correspond to return forces while rotational pose terms correspond to return moments:

$$\left(^{G}F_R\right)_i = -a_i \Delta X_i^2 ; \Delta X_i \leq \Delta X_{iDB}$$
$$= -a_i \Delta X_{iDB}^2 ; \Delta X_i > \Delta X_{iDB}$$

(16)

where the constant ΔX_{iDB} is the ith axis deadband value, a_i is the ith axis parabolic constant, and the subscript HC was dropped for clarity. If $\Delta X_i \leq \Delta X_{iDB}$, no Cartesian command is sent out for the ith axis. If $\Delta X_i > \Delta X_{iDB}$, ΔX_{iDB} must first be subtracted from ΔX_i before it is used in a Cartesian pose or rate command. Figure 11 and Equation 16 are represented on Figure 10 by the virtual spring characteristics K_R (more complicated than the other Figure 10 matrix gains due to the different zones). The rotational deadband should be applied at the angular velocity level due to the Euler angle coupling in rotation matrices.

The virtual parabolic walls displayed in Figure 11 are related to the virtual fixtures which have been applied in the past to the FREFLEX hardware (Rosenberg, 1992, 1993). The virtual walls assist an operator in separating axes for teleoperated rate inputs. The user's hand must overcome the virtual-wall force before inputs are sent on a certain axis. If the user's hand strays and no input is desired in certain Cartesian axes (translational or rotational), the virtual wall will return the operator's hand to the deadband area. On the other hand, virtual fixtures are used as a guide for the operator's hand in Cartesian space. For instance, if the user wishes to insert a peg in a specific hole, the virtual fixture can provide virtual springs (centered on the target hole) to assist guiding the peg to that hole.

To increase relative FRHC stability, a damping term is added. If the FRHC pose is static, there is zero damping term. However, if the operator is making FRHC pose changes with respect to time, the damping term applies a resistive wrench $^{G}F_D$ (opposite to the velocity direction of each Cartesian pose term) at the FRHC grip. This dampens rapid changes in the manipulator's Cartesian commands.

27

The $\Delta(\Delta X_{HC})$ vector is calculated via a simple difference in the current and previous ΔX_{HC} values. In this case we have small angle motion (for $\Delta[\Delta X_{HC}]$, not for ΔX_{HC}) so the entire pose representation ΔX_{HC} may be subtracted algebraically to yield $\Delta(\Delta X_{HC})$, rather than using the form of Equation 12. The vector $^{G}F_{D}$ is calculated by applying a diagonal matrix of damping gains K_{D} (with negative signs) to $\Delta(\Delta X_{HC})$:

$$\left(^{G}F_{D}\right)_{i} = -K_{Di}\Delta\left(\Delta X_{HC}\right)_{i}. \tag{17}$$

The stability issue is important in a wrench-reflecting system. If the operator makes contact between the manipulator and its environment at a high rate, a large wrench will be reflected, which pulls the operator's hand back. In turn, the manipulator will reverse, only to be returned with the command from the operator's hand recovering forward. This situation can lead to an oscillating instability. This is extremely difficult to model due to environment stiffness uncertainties and lack of a good model and variability for human operators. Stability can be aided by the FRHC damping term, but operator training and heuristic gain tuning also helped stabilize the system. A future goal is to better ensure stability.

To assist the operator the return-to-center and damping terms are summed to determine the required assist wrench $^{G}F_{RD}$ at the FRHC grip: $^{G}F_{RD}=^{G}F_{R}+^{G}F_{D}$. As with the task wrench case, $^{G}F_{RD}$ must be converted to the equivalent wrench for the Wrist Frame (Equation 13 with proper indices) before using an equation of the Equation 14 form to calculate the joint torques and forces τ_{RD} to achieve the assist features. In order to ensure that the task wrench dominates, the assist wrench τ_{RD} is first scaled uniformly to a given fraction of the FRHC joint torque limits to yield τ_{RDs}. The gravity compensation joint commands τ_{G} cannot be likewise scaled if they are to support the entire FRHC mass. The total assist joint torques are thus: $\tau_{RDG} = \tau_{RDs} + \tau_{G}$.

3.2.3 Total FRHC Joint Commands

The total joint torques and forces commanded to the FRHC joints are the sum of those required for the task wrench (with end effector and payload removed, if desired) and those required for the assist wrench: $\tau = \tau_{T} + \tau_{RDG}$. In order to calculate the final joint commands τ_{C}, a final uniform scaling (saturation) must be performed if one or more of the commanded joint torques τ exceed the motor capabilities. Also, gear ratios and torque calibration curves must be implemented (the torque mapping mentioned in Figure 9). Now the discussion of Figure 10 is complete. The operator feels any task wrenches and continuously updates the Cartesian manipulator commands, while being assisted by gravity compensation, return-to-center forces (for rate inputs), and FRHC damping forces.

3.2.4 Pose Versus Rate Cartesian Commands

The control architecture implemented allows both pose and rate inputs. Figure 10 is developed for the rate case, but the pose case is very similar. In pose mode, the CFRTC assist wrench is not required because a static FRHC pose yields a static manipulator pose (rather than moving with constant velocity). The deadband and virtual walls were first implemented at NASA but not found to be as useful as in the rate case (Williams, 1997). However, the same FRHC gravity compensation and damping terms apply well to the pose case.

CHAPTER 4 THE NATURALLY-TRANSITIONING RATE-TO-FORCE

CONTROLLER

Two fundamental problems in manipulator control are free motions in Cartesian space and contacting the environment during task performance. Resolved-rate control has been around for a long time (Whitney, 1969). However, rate control has not been widely implemented in practical industrial and remote operations, perhaps due to the difficulty of rate control in contact. If a constant rate is commanded while the manipulator contacts the environment, joint angles integrate until unacceptably large forces are exerted.

Compared to an inverse pose algorithm, the resolved-rate algorithm is attractive because it is a linearized, unique solution (assuming full rank for the Jacobian matrix). Also, control inputs from various sources can be summed linearly to form the total input command. Both inverse pose and inverse rate schemes are subject to the same singularities.

Raibert and Craig (1981) presented a hybrid control method wherein some Cartesian axes are controlled in position while the remaining axes are force controlled. While this method is effective in practical tasks, it does not use rate control, and one must choose either position or force on each Cartesian axis. Hogan (1985) presented an impedance controller where the behavior of a manipulator is controlled to mimic a six degrees of freedom Cartesian *m-c-k* system. Whitney (1985) reviewed various force control architectures.

Goldenberg, Apkarian, and Smith (1989) presented an approach which compensates for unknown loading and parameter uncertainty in computed torque manipulator control. Colbaugh, Seraji, and Glass (1993) presented an adaptive scheme for controlling the end-effector impedance of robot manipulators in contact; however, an explicit control mode change is required for free motion. Hyde and Cutkosky (1994) experimentally evaluated several methods for controlling the transition from free motion to constrained motion, using a one-axis impact testbed. Yao and Tomizuka (1995) presented an adaptive motion and force controller for manipulators with uncertainties in both the robot and contact surfaces. Vukobratovic and Stojic (1996) considered the problem of simultaneous stabilization of both the robot motion and interaction force in Cartesian space after contact in robotic tasks. Tarn, Wu, Xi, and Isidori (1996) used an event-driven switching control strategy for robot impact control and force regulation where the instant of impact is required.

This report presents a manipulator control method for effective task performance, the Naturally-Transitioning Rate-to-Force Controller. In free motion the manipulator moves with rate control, while in contact with the environment the force and moment wrench exerted on the environment is controlled. No artificial control mode or gain parameter changes are required so the transition is termed natural. A wrist-mounted force and torque sensor and force and moment accommodation algorithm are required. Rate and FMA are active on all Cartesian axes simultaneously so no hybrid scheme is necessary. Since there are no artificial mode changes required, the threshold of contact is unimportant.

The NTRFC was discovered through serendipity at NASA Langley Research Center (Williams et al., 1996). It was implemented at NASA experimentally and proven very effective in completion of representative space telerobotics tasks (Willshire et al., 1992). Though the NTRFC has shown great promise in the lab, its previous development has been exclusively heuristic. Therefore, the goal of this chapter is to demonstrate the NTRFC and provide methods for its design. Models and design methods are developed and evaluated. Considered are controller and manipulator dynamics, multiple degrees of freedom, transient and steady-state response, and stability, which were ignored in previous NTRFC work.

4.1 NTRFC Description

This section presents the Naturally-Transitioning Rate-to-Force Controller concept. It is applicable to control of any manipulator or manipulators equipped with wrist-mounted force and torque sensors, rate inputs, and contact with the environment. The system is presented for one manipulator, but dual-arm control has also been implemented (Williams et al., 1997).

Figure 12 shows the NTRFC high-level control diagram. The two basic active ingredients are the resolved rate and force and moment accommodation algorithms, described below.
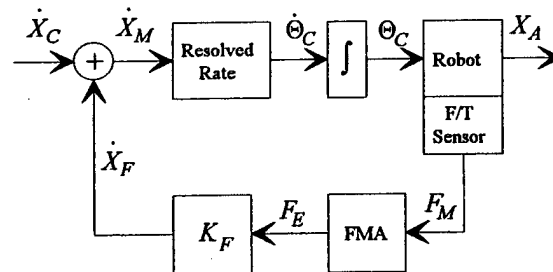
Figure 12. NTRFC Control Diagram

4.1.1 Force And Moment Accommodation

If the manipulator is in contact with its environment, there are constraints on X_A, which is the actual Cartesian pose in Figure 12, and a Cartesian wrench exists. An impedance controller (Hogan, 1985) with only the damping term has been implemented in the resolved-rate scheme to command forces to the environment with the manipulator. A six degree of freedom wrist-mounted force and torque sensor reads the current contact wrench $F_S = \{f_S \quad m_S\}^T$ expressed in the Sensor Frame. The weight and moment of the end effector mounted outboard of the sensor must be subtracted from the sensor reading, accounting for manipulator configuration. The modified sensor reading is transformed to the MRF wrench, F_M (Craig, 1989):

31

$$F_M = \begin{Bmatrix} f_M \\ m_M \end{Bmatrix} = \begin{bmatrix} {}^M_S R & 0 \\ {}^M P_S \times {}^M_S R & {}^M_S R \end{bmatrix} \begin{Bmatrix} f_S \\ m_S \end{Bmatrix}. \tag{18}$$

An error vector $F_E = F_C - F_M$ is formed from the difference of the sensed and commanded wrenches in the MRF and converted to a rate $\dot{X}_F = K_F F_E$, sent to the summing junction in Figure 12. This rate drives the manipulator motion so the desired force is achieved continuously. The diagonal gain matrix K_F has units m/Ns and rad/Nms for translational and rotational terms, respectively.

If a desired contact wrench F_C is commanded and the manipulator is in free motion, the rate input $\dot{X}_F$ will move the manipulator in the six degrees of freedom direction of F_C until the sensor senses F_C through contact with the environment. Then F_C is maintained without any controller changes.

If zero wrench is commanded ($F_C = 0$) and the manipulator is in free motion, $\dot{X}_F = 0$ (assuming a perfect force and torque sensor) because there is zero contact wrench F_M. If zero wrench is commanded and the manipulator contacts the environment, the $\dot{X}_F = -K_F F_M$ motion will automatically align the manipulator end effector for minimal Cartesian contact wrench and misalignments. This is called force and moment accommodation.

4.1.2 Naturally-Transitioning Rate-to-Force Controller

In the Naturally-Transitioning Rate-to-Force Controller, the resolved-rate algorithm acts simultaneously with the FMA algorithm (see Figure 12), for all Cartesian axes. Therefore no hybrid scheme is necessary. The overall resolved-rate input is the sum of the commanded rate and the FMA rate, $\dot{X}_M = \dot{X}_C + \dot{X}_F$. As the manipulator end effector approaches a wall in the environment, the rate controller commands motion through the wall, but the FMA controller commands a reverse motion to exert zero wrench. Therefore, an equilibrium condition is attained, where the rate input is proportional to the exerted Cartesian contact wrench. The NTRFC automatically corrects misalignments so insertion tasks can be completed with minimum contact wrenches. If no force controller is used, it is difficult to complete tasks since the manipulator is "blind" in the wrench sense.

The system behaves as a rate controller in free motion and as a force controller in contact. The transition requires no mode changes, logical switches, or gain changes in the controller software or hardware and thus is termed a natural transition. The transition is a consequence of the physics of manipulator contact with the environment when using the control architecture of Figure 12. Assuming a well-calibrated force and torque sensor with minimal noise, the NTRFC does not care when the moment of contact occurs. The FMA algorithm is enabled continuously and simultaneously with rate control on all Cartesian axes but only generates nonzero $\dot{X}_F$ in contact. The next section presents modeling and controller design for two manipulators with three degrees of freedom operating with the NTRFC .

32

4.2 NTRFC Modeling

This section presents dynamics and control modeling for a spatial $3P$ manipulator and a planar $3R$ manipulator in motion under the NTRFC. Since the free-motion to contact transition is a natural one, desirable performance must be obtained with only one set of gains and software control mode. Factors such as system stability, transient response, and steady-state response are important. Two control design procedures are presented, one in each of the following two subsections.

The NTRFC has been implemented heuristically in hardware at NASA Langley Research Center. The goal of this section is to provide an analytic, as opposed to heuristic, basis for NTRFC design. In hardware implementation the manipulator dynamics and environment characteristics are provided by the real world. In modeling, the control diagram in Figure 12 must be expanded to model these real world effects, as shown in Figure 13.

Figure 13 is the same as Figure 12, with the following added:

1. The Resolved Rate block is identified as the inverse Jacobian mapping, which is a function of the actual (modeled) joint angles Θ_A.

2. The commanded joint angles Θ_C are achieved using linear PID control for each joint independently.

3. The PID algorithms collectively yield the vector of input joint torques, from which the actual joint angles Θ_A are solved using forward dynamics, the manipulator equations of motion, and the contact wrench.

4. Both pose and rate forward kinematics are calculated to predict the current actual Cartesian pose X_A and rate which are not shown.

5. The environment model predicts the contact wrench F_M with the assumption of a perfect force and torque sensor. It is assumed that once contact with the environment is achieved, the manipulator and the environment remain in contact. In computer modeling of NTRFC motion, an artificial environment switch is used, but this is not necessary for hardware systems since the sensor reading is used continuously.

6. The FMA block is identified as the force error equation $F_E = F_C - F_M$. As in the Figure 12 case, FMA is continuously enabled, but only generates nonzero $\dot{X}_F$ when the manipulator is in contact with the environment.
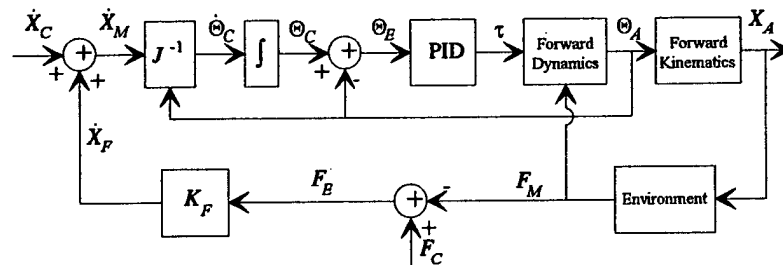
Figure 13. NTRFC Dynamics and Control Modeling Diagram

4.2.1 Spatial *3P* Manipulator

A three degrees of freedom spatial serial manipulator consisting of three orthogonal prismatic joints (P) is modeled and then the first NTRFC design procedure is presented. The *3P* diagram is presented in Figure 14 (front view, *YZ* plane) and Figure 15 (right side view, *XZ* plane). Prismatic actuator 1 and its ground connection are not shown in Figure 14 for clarity. The manipulator is modeled as three lumped masses m_i each with viscous dampers c_{Ai}, i = 1, 2, and 3. The relative compliance between the manipulator and the environment is modeled as three spring and damper combinations k_{Ej} and c_{Ej}, j = x, y, and z. Variable actuator lengths L_{A1}, L_{A2}, and L_{A3} operate along the X, Y, and Z axes, respectively. Fixed lengths L_{0j}, j = x, y, and z, give the distance along each axis from the origin to the undisplaced environment location. Cartesian variables x, y, and z are measured from the ends of L_{0j}.
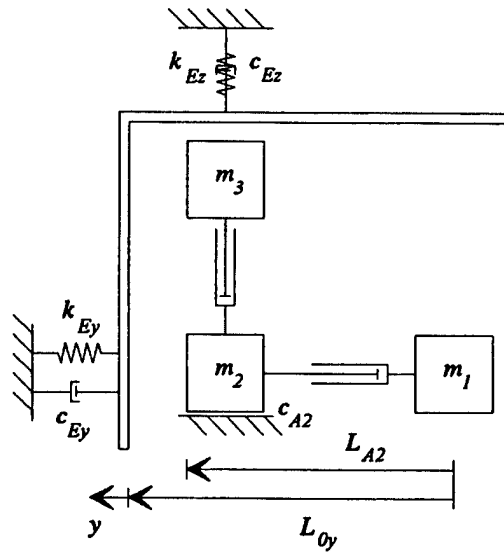
Figure 14. YZ Plane View of *3P* Manipulator Example
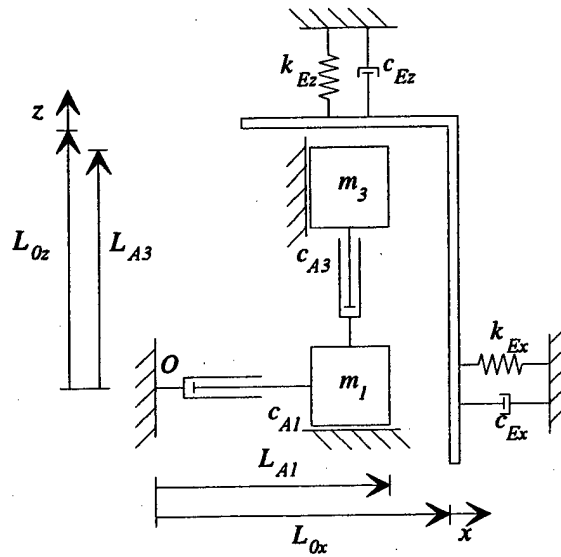
Figure 15. XZ Plane View of *3P* Manipulator Example

The *3P* manipulator has trivial resolved-rate ($J = J^{-1} = I_3$) and forward kinematics (L_{A1}, L_{A2}, and L_{A3} are the total displacements along X, Y, and Z) algorithms. No rotations are possible. The dynamics equations of motion are decoupled, each axis represented by

$$f_{Ai} + f_{Mj} = m_i \ddot{L}_{Ai} + c_{Ai} \dot{L}_{Ai} \qquad (19)$$

where i = 1, 2, and 3; f_{Ai} is the ith actuator force; f_{Mj} is the jth Cartesian MRF contact force; and $j = x, y,$ and z corresponds to i = 1, 2, and 3. The first actuator must accelerate m_1, m_2, and m_3. The second actuator accelerates m_2 and m_3. The third actuator just accelerates m_3. The reaction forces are resisted by the structure and not by other actuators because they are orthogonal. Therefore, the NTRFC design can proceed for each axis independently. Figure 16 shows joint 1 and only the X axis since Y and Z are similar.
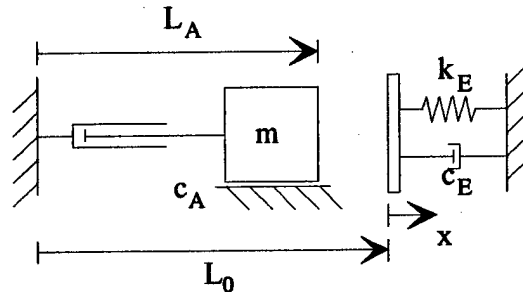
Figure 16. *3P* Manipulator Example Joint 1, X Axis

Given the 3*P* simplifications mentioned above, a linear single input and single output (SISO) diagram can be obtained for each joint by simplifying Figure 13, shown in Figure

35

17. In Figure 17, commanded rate $\dot{x}_C$ is the input. In free motion, L_A is the output, $-L_0$ is ignored, and f_M is zero so k_F has no effect since f_C is zero for FMA. In contact with the environment the input $\dot{x}_C$ no longer causes free motion but instead exerts a force on the environment. In this case f_M is the force and torque sensor reading, hence the force of the environment exerted back on the manipulator. The distance from the origin to the undisplaced environment, L_0, acts as a disturbance in this SISO system. The commonly used PID controller transfer function is

$$G_C = k_P + \frac{k_I}{s} + k_D s. \qquad (20)$$

The transfer functions for the manipulator joint dynamics and the environment are

$$G = \frac{1}{ms^2 + c_A s} \qquad (21)$$

and $\quad G_E = -c_E s - k_E. \qquad (22)$

Now NTRFC design is presented for the *3P* manipulator. Given desired contact force transient performance, calculate gains k_P, k_I, k_D, and k_F for the three axes.
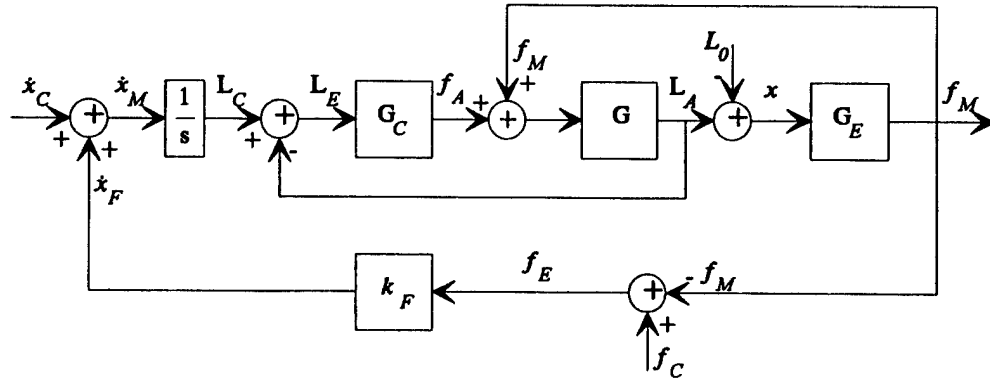
Figure 17. SISO Prismatic Joint NTRFC Block Diagram

This design procedure is for simple systems with decoupled kinematics and dynamics and linear models:

1. Derive the SISO transfer function for each independent axis and assume contact with the environment exists.

2. Set the desired stable transient performance characteristics of contact force given a rate step input.

3. Determine a desired fourth-order characteristic polynomial based on a dominant second-order system.

4. Using parameter matching, calculate PID and k_F gains for each joint separately and determine the stability ranges.

5. Ensure that the same PID gains yield acceptable performance under free motion (k_F is always enabled but has no effect until contact) and simulate the results.

The NTRFC design procedure for one P joint is now given. The linear superposition principle is used to find the total contact force output f_M given the rate input $\dot{x}_C$ with $L_0 = 0$ and the disturbance input $-L_0$ with $\dot{x}_C = 0$:

$$T_{\dot{X}} = \frac{f_{M1}}{\dot{x}_C}, \tag{23}$$

$$T_L = \frac{f_{M2}}{-L_0}, \tag{24}$$

$$\text{and } f_M = f_{M1} + f_{M2} = T_{\dot{X}}\dot{x}_C - T_L L_0. \tag{25}$$

The closed-loop transfer function $T_{\dot{X}}$ for Figure 17 under contact with $L_0 = 0$ is

$$T_{\dot{X}} = \frac{-\left[k_D c_E s^3 + \left(k_D k_E + k_P c_E\right)s^2 + \left(k_P k_E + k_I c_E\right)s + k_I k_E\right]}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}, \tag{26}$$

where $a_4 = m$,

$$a_3 = c_A + c_E + k_D - k_D c_E k_F, \tag{27}$$

$$a_2 = k_E + k_P - \left(k_D k_E + k_P c_E\right)k_F, \tag{28}$$

$$a_1 = k_I - \left(k_P k_E + k_I c_E\right)k_F, \tag{29}$$

$$\text{and} \quad a_0 = -k_I k_E k_F. \tag{30}$$

The closed-loop transfer function T_L for Figure 17 under contact with $\dot{x}_C = 0$ is different from Equation 26, but has the same characteristic polynomial. Therefore, design for transient response affects both superposition components in the same manner.

Given a step input $\dot{x}_C$, the two desired transient performance characteristics for f_M: 4% overshoot and 1 second settling time ($\pm2\%$) are set, which leads to $\xi = 0.72$ and $\omega_n = 5.59$ rad/s, for a dominant second-order characteristic polynomial $s^2 + 8s + 31.24$. This is a fourth-order characteristic polynomial with four unknowns. Therefore the dominant second-order polynomial is augmented with two negative real poles at least 10 times greater than the real part of the dominant poles to yield

$s^4 + 98s^3 + 2751s^2 + 18812s + 62482$. Therefore $a_4 = 1$, $a_3 = 98$, $a_2 = 2751$, $a_1 = 18812$, and $a_0 = 62482$. The corresponding poles are $s_{1,2,3,4} = -4 \pm 3.90i$, -40, -50.

Now parameter matching is used to derive an analytic solution for the four unknown gains. First, the desired fourth-order polynomial must be uniformly scaled so the leading coefficient is $a_4 = m$; then the remaining coefficient equations from Equation 26 are solved for the unknowns:

$$k_I = \frac{-a_0}{k_E k_F}, \tag{31}$$

$$k_D = \frac{a_3 - c_A - c_E}{1 - c_E k_F}, \tag{32}$$

$$\text{and } k_P = \frac{(a_0 c_E - a_1 k_E)k_F - a_0}{(k_E k_F)^2}. \tag{33}$$

The third order polynomial

$$b_3 k_F^3 + b_2 k_F^2 + b_1 k_F + b_0 = 0 \tag{34}$$

is solved for k_F. The coefficients

$$b_3 = (a_0 c_E - a_1 k_E)c_E^2 + a_2 c_E k_E^2 + (c_A - a_3)k_E^3, \tag{35}$$

$$b_2 = -a_0 c_E (1 + 2c_E) + 2a_1 c_E k_E - a_2 k_E^2 + k_E^3, \tag{36}$$

$$b_1 = 3a_0 c_E - a_1 k_E, \tag{37}$$

$$b_0 = -a_0, \tag{38}$$

and a_i ($i = 0, 1, 2,$ and 3) are the desired fourth-order polynomial coefficients (to force the desired behavior in the Equation 26 denominator). Since coefficients b_i are real, at least one real root is guaranteed. Choose the real k_F value and then Equations 31, 32, and 33 yield the unknowns k_I, k_D, and k_P given k_F. The steady-state contact force and environment displacements are found using the final value theorem (to predict these steady-state values and validate the simulation):

38

$$f_{M1_{ss}} = \lim_{t \to \infty} f_{M1}(t) = \lim_{s \to 0} s f_{M1}(s) = \lim_{s \to 0} s T_{\dot{X}} \frac{\dot{x}_C}{s} = \frac{\dot{x}_C}{k_F}$$

$$f_{M2_{ss}} = 0 \qquad f_{M_{ss}} = f_{M1_{ss}} + f_{M2_{ss}} = \frac{\dot{x}_C}{k_F} \qquad x_{ss} = \frac{-f_{M_{ss}}}{k_E} = \frac{-\dot{x}_C}{k_E k_F}. \qquad (39)$$

A *3P* manipulator simulation example is now presented. Apply the above design procedure using Equations 31, 32, 33, and 34 three times independently, one for each *P* joint. Given the following parameters in Table 5, solve for k_{Pi}, k_{Ii}, k_{Di}, and k_{Fj} ($i = 1, 2,$ and $3; j = x, y,$ and z) and simulate NTRFC motion. Note that i represents joint space and j Cartesian space, identical for the *3P*.

Table 5. *3P* Manipulator Parameters

Mass (kg)	Damping (kg/s)	Length (m)	Rate input (m/s)	Environment stiffness (kg/s²)	Environment damping (kg/s)
$m_1 = 3$	$c_{A1} = 0.3$	$L_{0x} = 0.20$	$\dot{x}_C = 0.1$	$k_{Ex} = 100$	$c_{Ex} = 0.7$
$m_2 = 2$	$c_{A2} = 0.5$	$L_{0y} = 0.10$	$\dot{y}_C = 0.2$	$k_{Ey} = 100$	$c_{Ey} = 0.7$
$m_3 = 1$	$c_{A3} = 0.4$	$L_{0z} = 0.12$	$\dot{z}_C = 0.3$	$k_{Ez} = 100$	$c_{Ez} = 0.7$

Table 6 summarizes the *3P* solution for NTRFC design. For joint 1 $m = m_1 + m_2 + m_3$, $m = m_2 + m_3$ for joint 2, and $m = m_3$ for joint 3. The desired characteristic polynomial for each case (derived above) is $s^4 + 98s^3 + 2751s^2 + 18812s + 62482$, uniformly scaled so the leading coefficient is $a_4 = m$ for each joint.

Table 6. *3P* NTRFC Design Results

Gain	Axis 1	Axis 2	Axis 3
k_P	4063.0	2050.6	711.5
k_I	16197.0	8190.6	2867.2
k_D	505.1	252.4	84.1
k_F	-0.232	-0.229	-0.218

A MATLAB SIMULINK model was developed to simulate the *3P* under NTRFC motion. In this simulation, the inputs $\dot{x}_C$, $\dot{y}_C$, and $\dot{z}_C$ were each ramped up to their final value with a slope of one. As shown in Figure 18, the actuator lengths (also the global Cartesian displacements) each increase linearly under rate control in free motion, experience transient behavior (difficult to see at this scale), and assume their steady-state value after the natural transition from rate to force control. With the arbitrary simulation values chosen, each axis moves with different rates and the time of contact with the environment is different. The simulated MRF contact forces in Figure 19 show these different times of contact. Also, though difficult to see at this scale, each force transient

behavior satisfies the desired 4% overshoot and 1 second settling time. These control goals are not met exactly due to the fourth-order approximation of the dominant second-order polynomial and the PID controller adds zeros to the system. Figure 19 shows that the contact forces are zero in free motion until each axis contacts the environment; they also experience transient behavior and assume steady-state (constant) force values after the transition even though the rate commands are still applied. Figure 20 shows the actuator forces required to achieve NTRFC motion. To initiate the constant-rate free motion, each force briefly approximates a step input. During the transition from rate to force control, each axis requires a sharp change in actuator force. The actuator force f_{A3} has a static offset $m_3 g$ to resist gravity loading.
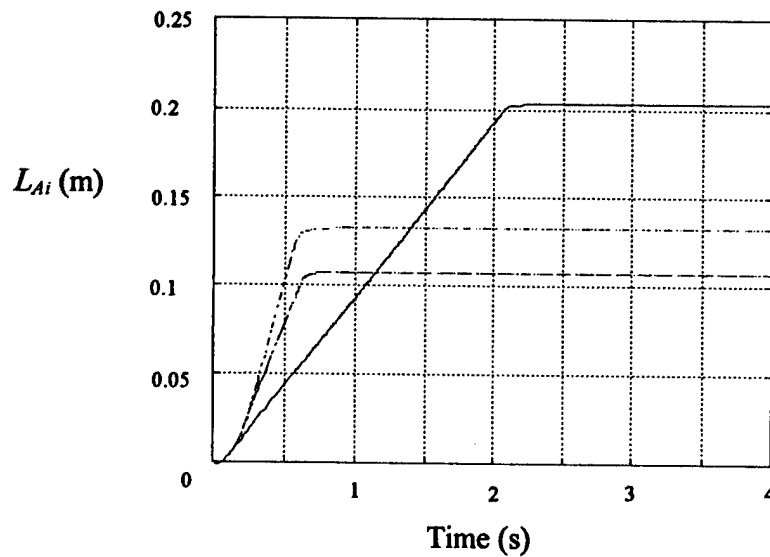
Figure 18. Actuator lengths of *3P* manipulator versus time in seconds. Note L_{A1} is solid, L_{A2} is dashed, and L_{A3} is dash-dotted.
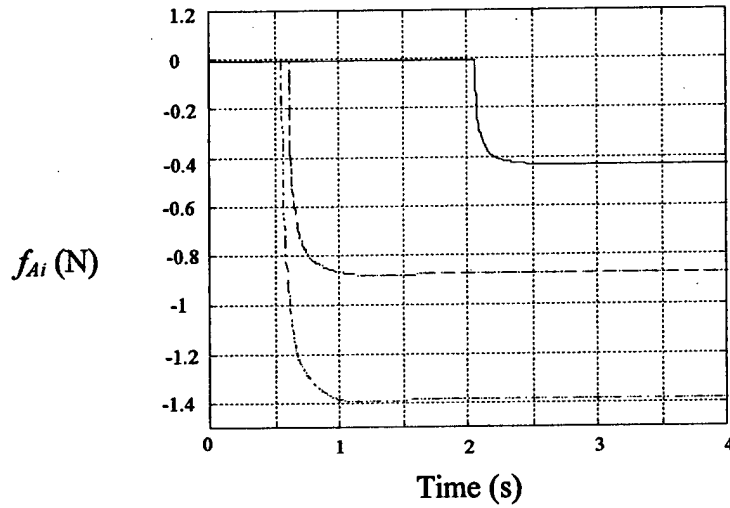
Figure 19. Contact forces of *3P* manipulator versus time in seconds. Note f_{Mx} is solid, f_{My} is dashed, and f_{Mz} is dash-dotted.
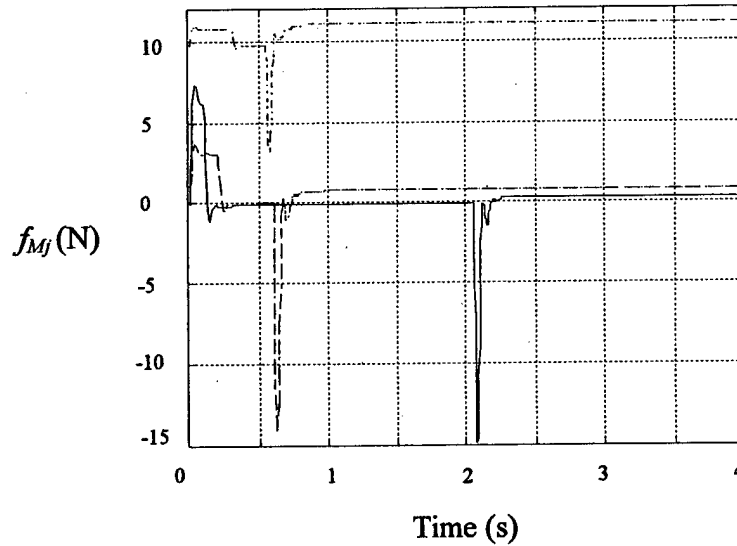
Figure 20. Actuator forces of *3P* manipulator versus time in seconds. Note f_{A1} is solid, f_{A2} is dashed, and f_{A3} is dash-dotted.

The steady-state values calculated from Equation 39 are $f_{Mx_{SS}} = -0.432$ N, $f_{My_{SS}} = -0.874$ N, $f_{Mz_{SS}} = -1.377$ N, $x_{SS} = 0.00432$ m, $y_{SS} = 0.00874$ m, and $z_{SS} = 0.01377$ m.

Given the PID gain values from Table 6, the gains k_{Fj} for each Cartesian axis were varied to investigate stability. The stability results shown in Figure 21 are identical for each axis since the same characteristic polynomial (albeit scaled) was used for all three axes. Figure 21 reports the real part of the four poles for each axis. The full behavior is

41

adequately represented by the range $-1 \le k_{Fj} \le 1$. The two complex conjugate poles (dash) have a relatively small negative real part over the entire k_{Fj} range. The two real poles are negative and identical up to k_{Fj} equals -0.24 (close to the design values for each k_{Fj} in Table 6). After this point the real poles bifurcate. One becomes more negative. The other becomes zero at k_{Fj} equals 0 and positive for positive k_{Fj}. Therefore, the 3P system under NTRFC motion is marginally stable for k_{Fj} equals 0 and unstable for k_{Fj} greater than 0. This result makes sense physically since k_{Fj} greater than 0 would cause a rate to increase the contact force on axis j.
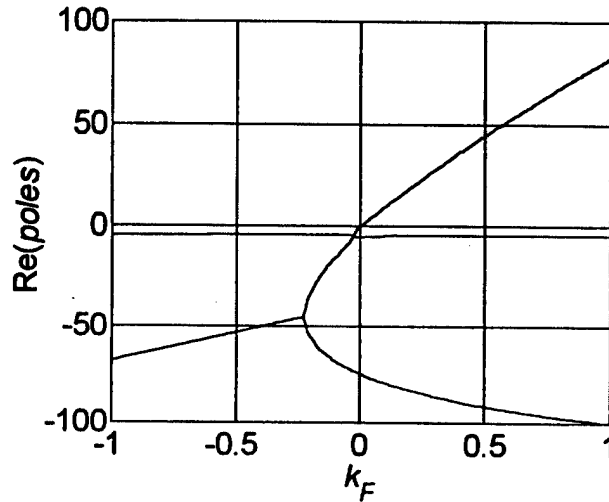
Figure 21. Stability results of *3P* manipulator shown as a plot of the real poles versus k_{Fj}.

The gains in Table 6 were designed for the contact case; now the free motion characteristics must be analyzed using the same gains. The motion looks fine in the free motion portions of Figure 18, but the performance is quantified below. Referring to Figure 17 and using G_C and G from Equations 20 and 21, the transfer function representing PID control and joint dynamics is

$$T_{L_A L_C} = \frac{L_A}{L_C} = \frac{k_D s^2 + k_P s + k_I}{ms^3 + (c_A + k_D)s^2 + k_P s + k_I} \quad . \tag{40}$$

The three free-motion poles for all axes are nearly identical (-4.23±4.21i and -75.8, -4.27±4.22i and -75.8, and -4.46±4.25i and -75.6 for joints 1, 2, and 3) and provide 7.9% overshoot and 0.26 s settling time for output L_{Ai} with input L_{Ci}. All axes are stable for the designed PID gains since the poles all have strictly negative real parts.

4.2.2 Planar *3R* Manipulator

A three degrees of freedom planar serial manipulator consisting of three parallel revolute joints (*R*) is modeled and the second NTRFC design procedure is presented in this section. The *3R* diagram is shown in Figure 22. The manipulator is modeled as three

distributed masses m_i with inertia scalars I_i, $i = 1, 2,$ and 3. The relative manipulator and environment compliance is modeled as three spring and damper combinations k_{Ej} and c_{Ej}, where $j = x, y,$ and r (two translational and one rotational). The fixed lengths are $L_1, L_2,$ and L_3. The variable joint angles $\Theta_A = \{\theta_1 \quad \theta_2 \quad \theta_3\}^T$ are controlled by joint torques $\tau = \{\tau_1 \quad \tau_2 \quad \tau_3\}^T$. The rate inputs are $\dot{X}_C = \{\dot{x}_C \quad \dot{y}_C \quad \omega_z\}^T$, relative to MRF ($X_M$ is aligned with link 3). The Cartesian pose is $X_A = \{x \quad y \quad \phi\}^T$ and the Cartesian contact wrench is $F_M = \{f_x \quad f_y \quad m_z\}^T$. The 3R forward kinematics solution and Jacobian matrix reflect joint coupling and are straight-forward to derive. The three dynamics equations of motion are coupled and nonlinear, represented by

$$\tau = M(\Theta_A)\ddot{\Theta}_A + V(\Theta_A, \dot{\Theta}_A) + G(\Theta_A). \qquad (41)$$

The terms in Equation 41 are rather complex, but straight-forward to derive (Craig, 1989). The 3R system is coupled and nonlinear so the procedure presented for the 3P case discussed earlier cannot be applied. Figure 22 represents the 3R case. Now the Second NTRFC Design Procedure is presented. The goal is still to calculate k_{Pi}, k_{Ii}, k_{Di} for the three R joints and k_{Fj} for the three Cartesian directions.
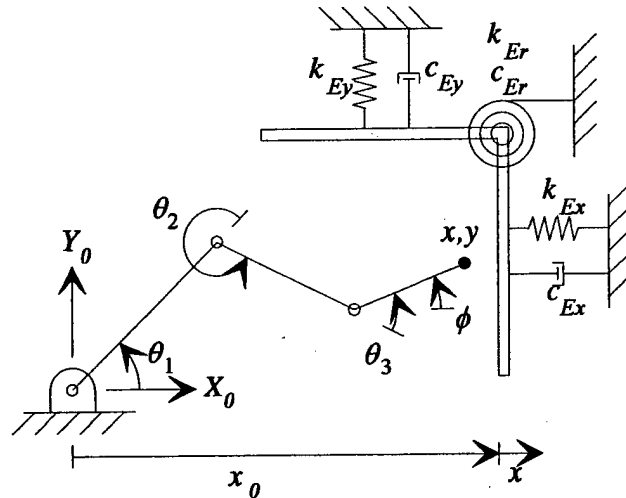
Figure 22. 3R Manipulator

The 3R NTRFC Design Procedure is now presented. This method should be used for more complex systems with coupled kinematics and dynamics and nonlinear models. Set the desired (stable) transient performance characteristics of each joint angle given a rate step input. Employ standard methods to design PID gains independently for each axis assuming decoupled dynamics and free motion. Using the same PID gains, choose a diagonal matrix gain K_F to ensure stability and to satisfy the desired contact wrench

performance for the given rate step inputs. Determine the stability ranges and simulate the results.

A *3R* manipulator simulation example is now presented. Given the parameters in Table 7, solve for k_{Pi}, k_{Ii}, and k_{Di} ($i = 1$, 2, and 3), determine K_F, and simulate NTRFC motion.

Table 7. *3R* Manipulator Parameters

Link length (m)	Link mass (kg)	Link moments of inertia (kg m²)	Environmental stiffness (kg/s²)	Environmental damping (kg/s)
$L_1 = 0.3$	$m_1 = 2$	$I_1 = 0.5875$	$k_{Ex} = 100$	$c_{Ex} = 0.7$
$L_2 = 0.2$	$m_2 = 1.5$	$I_2 = 0.0775$	$k_{Ey} = 100$	$c_{Ey} = 0.7$
$L_3 = 0.1$	$m_3 = 1$	$I_3 = 0.0025$	$k_{Ez} = 50$	$c_{Ez} = 0$

Assuming decoupled dynamics and free motion, the PID gains are determined independently for each joint. The transfer function relating actual joint angle to commanded joint angle is T_θ, where the decoupled linearized joint dynamics plant G was used:

$$T_\theta = \frac{\theta_A}{\theta_C} = \frac{k_D s^2 + k_P s + k_I}{I s^3 + (c + k_D)s^2 + k_P s + k_I} \qquad G = \frac{1}{I s^2 + cs}. \qquad (42)$$

The viscous damping coefficient, $c = 0.2$, is included for each joint. Now we specify 3% overshoot and 0.5 second settling time for each θ_A in free motion, yielding a desired characteristic polynomial (dominant second order augmented with real third pole ten times greater) $s^3 + 96s^2 + 1395s + 9230$. This polynomial must be scaled for each joint so $a_3 = I_i$ and then k_{Pi}, k_{Ii}, and k_{Di} are calculated via parameter matching with the denominator of T_θ in Equation 42. The results are shown in Table 8.

Table 8. *3R* Free Motion PID Design Results

Gain	Axis 1	Axis 2	Axis 3
k_P	819.8	108.1	3.5
k_I	5422.4	715.3	23.1
k_D	56.2	7.2	0.04

A MATLAB SIMULINK model was developed to determine FMA diagonal gain matrix K_F and simulate the *3R* under NTRFC motion. In this example, the *3R* manipulator is to make contact with the environment by moving along the horizontal X axis at a constant rate $\dot{X}_C = \{0.05 \ 0 \ 0\}^T$; this could simulate a peg-in-the-hole task. However, assume there is an unknown and undesired angular misalignment: Given initial angles $\Theta_{A_{in}} = \{45 \ -90 \ 65\}^T$, the forward kinematics solution yields the initial Cartesian pose $X_{A_{in}} = \{0.448 \ 0.105 \ 20\}^T$ (similar to Figure 22 pose), with an angular misalignment $\phi = 20$.

Therefore, when commanding $\dot{X}_C$ in MRF coordinates, the resulting motion will not be horizontal, but along link 3, inclined at 20 degrees. Free motion will continue until the environment has been contacted in the X_0 direction (at $x_0 = 0.50$, see Figure 22). Assume perfect contact in all Cartesian directions from that point in time forward. Then the equilibrium point for Y environment motion is $y_0 = 0.124$, which is the point of initial contact, but the equilibrium angle is $\phi_0 = 0$, which represents an immediate angular misalignment upon contact. The point of this simulation is to demonstrate how the angular misalignment will automatically correct itself under NTRFC motion.

Using the Table 8 free motion PID gains and trial-and-error with SIMULINK, a "good" value for K_F in contact was found to be a diagonal matrix of dimension 3 with a value of -0.1 for all three diagonal elements. "Good" is defined to be stable with reasonable transient performance. Figures 23 through 25 show simulated results for this example. Note in Figures 24 and 25 that the angular terms have separate scales on the right. Figure 23 shows the simulated joints angles Θ_A. As shown in Figure 24, the Cartesian pose variables x and y in $\{0\}$ coordinates each increase linearly under rate control in free motion, experience transient behavior (difficult to see at this scale), and assume their steady-state value after the natural transition from rate to force control. Since there is only an X rate command, the x steady-state value compresses the environment, beyond $x_0 = 0.50$, but the y steady-state value settles at $y_0 = 0.124$. Due to simulated forward dynamics, ϕ briefly exceeds the misalignment 20 degrees but then maintains that value in free motion. Upon contact, the NTRFC drives the manipulator to rectify the angle misalignment, sending ϕ to $\phi_0 = 0$. Figure 25 shows the simulated Cartesian contact wrench. The component f_x behaves similarly to any axis of the $3P$ case (see Figure 19), except there is a more interesting transient due to coupled dynamics. The component f_y also starts at zero in free motion, experiences a transient which gradually increases from zero, and then settles down to zero since $\dot{y}_C = 0$. The contact moment m_z is also zero in free motion, but experiences a step change on contact due to the angle misalignment. After the transient, the steady-state moment is zero since there is no ω_z rate command. When the moment step change occurs, the rotational term of the rate input $\dot{X}_F$ from FMA is no longer zero, but drives the manipulator in the direction to relieve the moment and hence the angular misalignment. When the angular misalignment has been eliminated, the MRF coordinates line up with $\{0\}$ as desired.
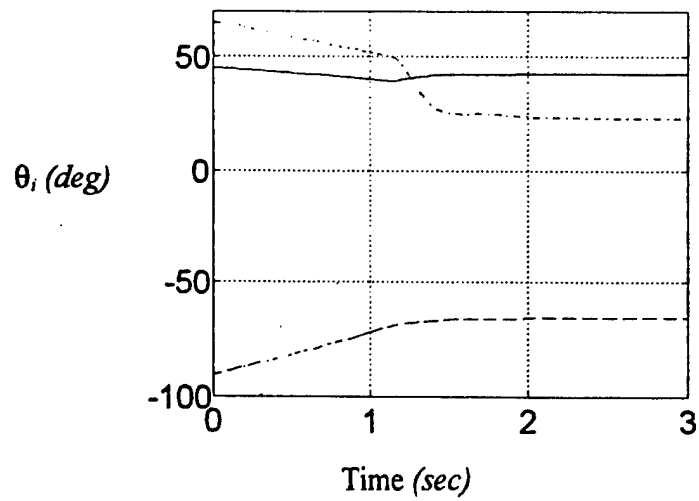
Figure 23. Joint Angles of *3R* manipulator versus time in seconds. Note that θ_1 is solid, θ_2 is dashed, and θ_3 is dash-dotted.
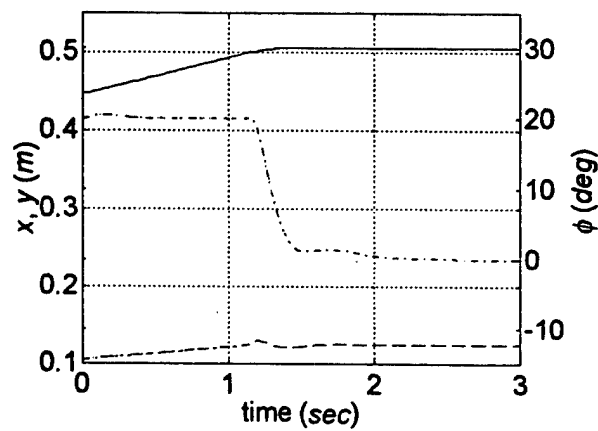

Figure 24. Cartesian Pose of *3R* manipulator versus time. Note *x* is solid, *y* is dashed, and ϕ is dash-dotted. The scale for phi is the right dependent axis and is labeled in degrees.
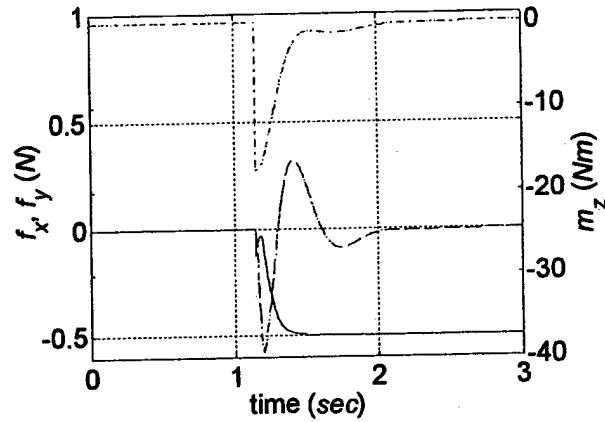
Figure 25. Contact Wrench of *3R* manipulator. Note f_x is solid, f_y is dashed, and m_z is dash-dotted. The scale for m_z is the right dependent axis and is labeled in Nm.

Even though the *3R* system is coupled and nonlinear, the steady-state values can be calculated using Equation 39. The result is $f_{x_{SS}} = -0.50$, $\Delta x_{SS} = 0.0050$, $f_{y_{SS}} = m_{z_{SS}} = \Delta y_{SS} = \Delta \phi_{SS} = 0$. Stability analysis was conducted by SIMULINK simulation. It was found that the *3R* has the same stability conditions as the *3P* case: marginally stable for any one $k_{Fj} = 0$ and unstable for any one $k_{Fj} > 0$. However, perhaps due to the joint coupling, the unstable behavior was different for the two cases. The *3P* suffers exponential increases in X_{Aj} and f_{Mj} whenever $k_{Fj} > 0$. For the *3R*, the unbounded outputs increase only linearly (with oscillations about the line) whenever any one $k_{Fj} > 0$.

Behavior varies widely for other K_F values. Unstable cases are discussed in the previous paragraph. For K_F values negative and larger than the design results reported, the systems are stable and achieve steady-state values as calculated in Equation 39. However, significant and unacceptable transient oscillations can occur. For K_F values negative and smaller than the design results reported, the systems are stable but sluggish to reach steady-state. Finally, when all $k_{Fj} = 0$, our stability conclusions predict marginally stable systems. However, this case corresponds to turning off the FMA algorithm and hence no natural transition from rate to force control occurs. Given a constant rate command $\dot{x}_c$, the manipulator performs fine in free motion, but without FMA generates unacceptably high forces in contact. With properly-designed K_F, the NTRFC provides excellent contact characteristics with a rate controller and no mode changes.

CHAPTER 5 MATLAB KINEMATIC SIMULATION AND HSF HARDWARE

IMPLEMENTATION

This chapter briefly describes the MATLAB simulation used to develop the control algorithms and the implementation of this control method to the HSF hardware.

5.1 MATLAB Kinematic Simulation

A MATLAB simulation was developed for the FREFLEX and MERLIN system using the control architecture of this report. This simulation is useful to validate algorithms, test new ideas safely, compare data from hardware implementation, and view simulated motions. This code was developed under MATLAB 5.0 but is compatible with previous versions. This section presents the required m-files and their hierarchy. Joint, pose, and rate control modes are available. The menu-driven simulation is invoked by typing "fremer" at the MATLAB prompt. Table 9 lists and describes the MATLAB simulation m-files.

Table 9. MATLAB Simulation M-files

M-file	Description
fremer.m	Main FREFLEX and MERLIN routine; calls other m-files
merlinDH.m	MERLIN DH parameters and other constants
freDH.m	FREFLEX DH parameters and other constants
merfk.m	MERLIN forward kinematics including animation vectors
frefk.m	FREFLEX forward kinematics including animation vectors
Fmplot.m	Display FREFLEX and MERLIN in current poses (animation)
fretrq.m	Calculate FREFLEX force and torque algorithms; calls Fj0.m and fregc.m
frein.m	Simulate user input on FREFLEX, apply deadbands
Fpose.m	Move FREFLEX to user-specified pose; calls Frr.m
Mpose.m	Move MERLIN to user-specified pose using resolved-rate; calls Mrr.m
Mpose2.m	Move MERLIN to user-specified pose using inverse pose; calls smoothie.m
plotres.m	Plot simulation results, if user desires
Fj0.m	Calculate FREFLEX Jacobian and additional kinematics
fregc.m	Calculate FREFLEX gravity compensation
Frr.m	FREFLEX resolved-rate, only for simulation Fpose.m
Mrr.m	MERLIN Jacobian and resolved-rate
smoothie.m	MERLIN joint angle interpolation, inverse pose only
dhfun.m	Evaluate Equation 3 given one row of DH parameters
inv_homo.m	Invert homogeneous transformation matrix (Craig, 1989)
inv_euler.m	Extract ZYX Euler angles from a rotation matrix, Equation 11

Figure 26 shows a four-view graphical snapshot of the FREFLEX and MERLIN system during a typical resolved-rate motion. The FREFLEX held at a Cartesian pose

away from its reference pose G_0 causes the MERLIN to move with a corresponding six degrees of freedom Cartesian rate.
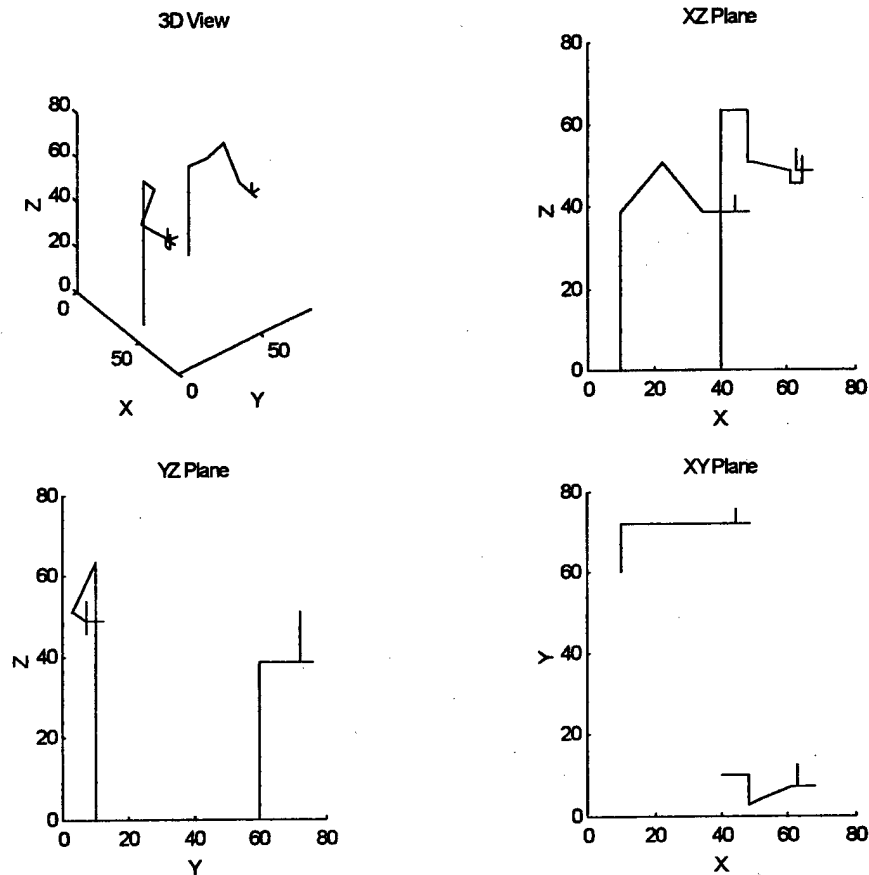
Figure 26. Fremer Simulation Graphics

5.2 HSF Implementation

This section briefly describes the implementation issues and some operational guidelines for the control methods installed for MERLIN teleoperation via the FREFLEX in the HSF Laboratory.

5.2.1 Implementation Issues

Issues that needed to be resolved for successful implementation of the control architecture into the FREFLEX and MERLIN system are discussed in this section.

A workspace singularity exists when joint 5 of the MERLIN is at zero degrees. When the wrist is in this position, joint axes 4 and 6 line up resulting in a loss of one degree of freedom.

While operating in resolved-rate mode, the Jacobian is inverted using Gaussian elimination to determine the proper joint rates. Therefore when joint 5 passes through this singularity, the robot wrist tends to rotate wildly for a few seconds before settling at a position outside the singular region. Singular Value Decomposition (SVD) with the damped least-squares solution can be utilized to avoid this loss of control. This method will return a useful but not necessarily expected solution. If the Jacobian is singular, SVD with the damped least-squares solution can be used to find a suitable inverse which will result in feasible joint rates, minimize the MRF trajectory error, and allow the operator to maintain control of the robot (Maciejewski & Klein, 1989).

A SVD program was developed in C on a PC for easy debugging before implementation in the Chimera system. This program produced valid solutions. However when the program was moved to the Chimera system operating on the Sun, the program did not yield valid solutions. The program was tested on the Sun using two compilers but the solutions were not satisfactory. The program was tested on several personal computers and it ran successfully on each.

The PC and Sun programs were modified to display the result for each line of code and these results were compared. The point at which the programs begin to diverge was discovered. The line of code that was causing the differing results was a simple division. The denominator in this division was very small (approximately 10^{-15}). Therefore the systems most likely have different levels of accuracy. A solution to this problem was not found because of time constraints and the singularity has been avoided by performing tasks that do not require passage through the singular region.

The CFRTC force was implemented and operates effectively except for certain wrist rotations. For example if the wrist is yawed in one direction, a return-to-center moment is exerted on the hand to counteract this orientation. However if the wrist is yawed in the opposite direction, the return-to-center moment is not perceived. This inconsistent moment reflection was also observed when only gravity compensation torques were being produced by the FREFLEX. This might result from an inaccurate FREFLEX joint decoupling matrix or improper summing of torques to be reflected and should be further studied.

Stability was not a problem with FREFLEX teleoperation and therefore the damping auxiliary force was not fully implemented. It has been partially installed but has not been thoroughly tested or debugged.

Pose control utilizing the resolved-rate algorithm was not successfully implemented into the HSF hardware. The program did not converge within the period required by the real-time operating system. The program might converge within the allotted time if the velocity tolerances were increased but a large error would result between the desired and actual position of the manipulator. Therefore only pose control using inverse position kinematics is currently installed in the HSF Lab. The pose method using the resolved-rate algorithm could be operational in the future if a faster processor was utilized or more efficient use of the Chimera modules was achieved.

Initial force-reflection values received from the force sensor mounted at the wrist of the MERLIN caused the FREFLEX motors to deactivate. Experiments showed that a large change in torque and not a maximum torque value will activate this FREFLEX safety

mechanism. The force reflection gains were reduced to a level at which the user can perceive the forces and the FREFLEX motors are unlikely to shut down due to large forces being sent from the MERLIN force sensor.

The MERLIN wrist-mounted JR^3 sensor was the source of most of the unexpected problems experienced during implementation. Incorrect readings were caused by vibrational noise during MERLIN motion and a *floating* ground within the building electrical system. These erroneous readings cause improper force reflection to the operator via the FREFLEX or inappropriate MERLIN motion under force control.

Activation of the FREFLEX motors for gravity compensation was causing an increase in the JR^3 force and moment values. The FREFLEX motors draw a large amount of current when activated. Since the ground wire has a finite resistance, this current causes a slight change in potential on the ground wire. The sensor electronics enclosure amplifies signals from the sensor on the magnitude of only 10^{-3} Volts. Therefore a minute shift in ground potential can be perceived by the sensor. The change in these values vary during different physical system configurations and these arrangements were studied to determine an acceptable setup. Figure 27 shows a plot of typical sensor noise.
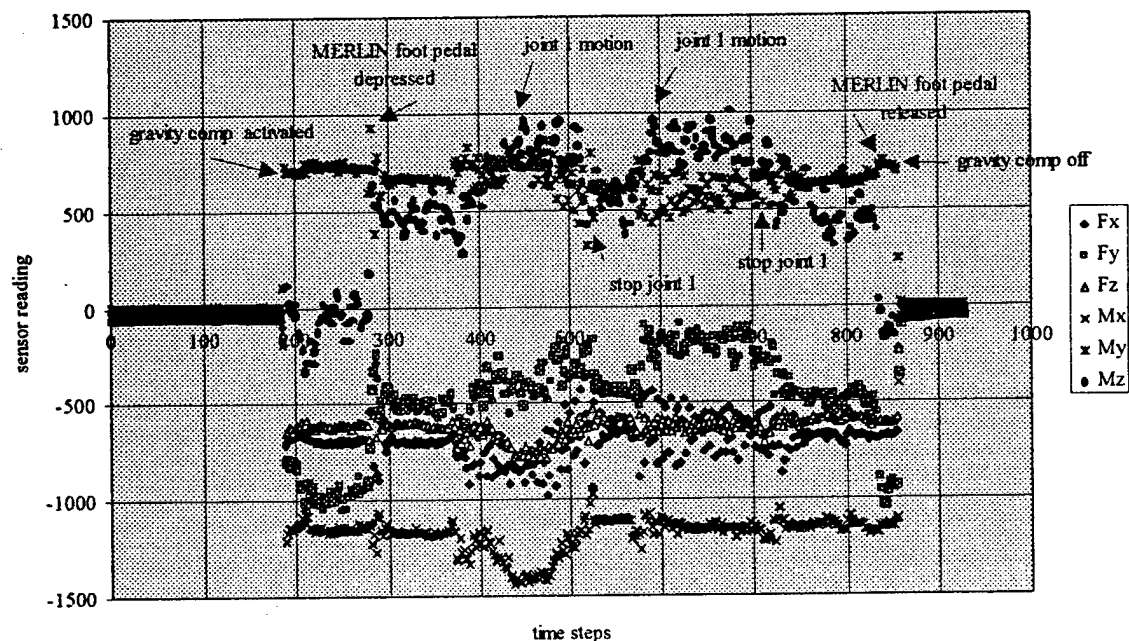
Figure 27. Typical JR^3 Sensor Noise

Sometimes the increase in the sensor values due to noise would be as much as 10 percent of the full-scale loading ($\pm$16384) for the sensor.

The arrangement that best minimizes the shift in ground potential includes two ground wires connecting the FREFLEX chassis to the MERLIN base; a three wire serial line that only has transmit, receive, and ground wires connecting the sensor electronics enclosure to the serial port labeled *zero* on the real-time processor Control; and a cable, which is split from the serial line that connects the sensor electronic enclosure to the serial

51

port on Control, connecting the sensor electronic enclosure to a serial port on the PC labeled "Spooky 2". Plugging the sensor into the electrical outlet of circuit 1 located beside the breaker panel also improves the performance.

The vibrational noise that is apparent in Figure 27 was minimized by adjusting the sensor timebase, digital filters, and notch filters. The current notch filter values are stored under C:\ETC and in file config0.ba on the MERLIN PC which is within the control cabinet. The original values are located in the file config0.jeb.

The digital filters are low-pass filters. Filter number 3 provides the highest amount of filtration (JR³, Inc., 1988). This filter eliminates noise that has a frequency faster than 12.5 % of the sensor timebase. Therefore to optimize this filter, the timebase for the JR³ was reduced to the smallest possible value which is 35 Hz. Filter 3 then eliminates noise frequencies higher than approximately 4.2 Hz. The timebase and the filter are set in the module jr3_serial.c.

According to JR³ (1988), the sensor has a right-hand coordinate frame with origin at the center of the sensor. However, real-world raw data from the sensor indicates that the Z axis is reversed and therefore the sensor measures forces with respect to a left-hand coordinate system with its origin located at the center of the sensor. The orientation of the theoretical and actual sensor frames is shown in Figure 28.
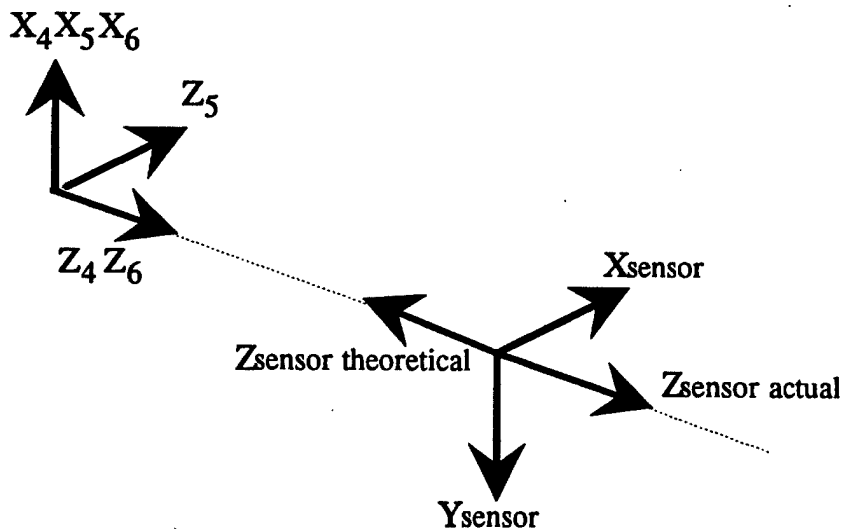
Figure 28. MERLIN Wrist-Mounted JR³ Sensor Frame Orientation

The raw data from the sensor is converted to lbf or in*lbf and the force in the direction of the Z axis and the moment about the Z axis are reversed immediately in the module force_ref.c. This transforms the sensor readings to the preferred right-hand coordinate system before these values are used in calculations.

The weight of any item mounted outboard of the sensor such as a gripper or other tool must be removed from the sensor readings. This weight causes the JR³ to sense forces and moments that vary with orientation even though no external loads are being applied to the system. The removal of this outboard weight allows the JR³ to correctly sense contact forces and proper force reflection and force control are possible.

The JR^3 removes the offsets by default for the sensor orientation at initialization. However this offset removal does not compensate for the outboard weight in different orientations. Therefore this command was removed so that the sensor would not "zero" its readings upon initialization. The command to not "zero" the sensor readings upon initialization is sent to the sensor in jr3_serial.c.

A program outbrd_wght.c was created to eliminate the weight outboard from the sensor readings in all orientations. This program calculates the mass outboard of the sensor and its center of gravity.

A notable delay exists from the instance a force is exerted on the JR^3 to the time the force is reflected to the user by the FREFLEX or observed as motion of the MERLIN under force control. According to Dr. D.W. Repperger (personal communication, January 9, 1998), a delay as small as 70 ms can be perceived by a human operator. This can reduce the advantages of force reflection in teleoperation and even hinder the operators performance if the delay is too large. Sometimes the operator may compensate for the delay and thus slow the operation.

Operating in NTRFC mode with large delays can result in large contact forces since the user is expecting the robot to react once contact is made. If the MRF collides with an object and the user continues commanding the robot to push forward, delays will not allow the robot to react fast enough to the contact forces. Rate control might yield lower contact forces under these circumstances since the user will know to stop commanding rates upon environmental contact.

Brief experiments were performed to obtain the approximate values of the delays. The time delay between the moment of contact with the environment and the reflection of this force to the teleoperator through the FREFLEX averaged approximately 230 ms. The time delay between the moment of contact and the response of the MERLIN during force accommodation averaged approximately 700 ms. Both of these delays exceed the desired response time.

The source of these delays has not been positively identified. Since delays are only noticeable while using the MERLIN mounted JR^3 sensor, the serial connection between the sensor and the processing board Control is the primary suspect for these delays. The signals from the sensor are being sent to the processing board Control through a 9600 baud serial line. Each character is composed of 12 bits and a complete frame of data from the JR^3 is approximately 20 characters long. Dividing the baud rate by 12 yields 800, the characters per second being sent through the serial line. Dividing this number by 20, the number of characters in a data frame, yields 40 which is the number of complete data strings per second being sent to the processor Control. The program jr3_serial.c then reads 40 characters per cycle to ensure that a complete data string is found. Therefore the fastest frequency that jr3_serial can sustain is approximately 20 Hz. The frequency for sending commands to the MERLIN via merliniod was determined by the same method to be 15 Hz.

A maximum time delay of 50 ms should result when jr3_serial operates at 20 Hz. This is still below the human perception threshold mentioned above by Dr. Repperger and much lower than the measured delay. Attempts have been made to contact the producers of Chimera but no response has been received. The validity of the JR^3 sensor operating

frequency and the Chimera module frequencies is questionable because of this unaccountable delay.

Initial attempts to command the MERLIN by sending joint angles using the HSHI were not successful. The motion of the robot joints was not as expected. This problem was corrected by performing the joint angle to encoder tick conversions in merliniod as listed by American Robot Corporation (1986).

5.2.2 General Operation

This section discusses some important points of basic operation within the current system. See Appendix A for complete operating procedures. The basic Chimera module flow diagram for FREFLEX teleoperation of the MERLIN is shown below in Figure 29. See Appendix B for descriptions of each Chimera module.
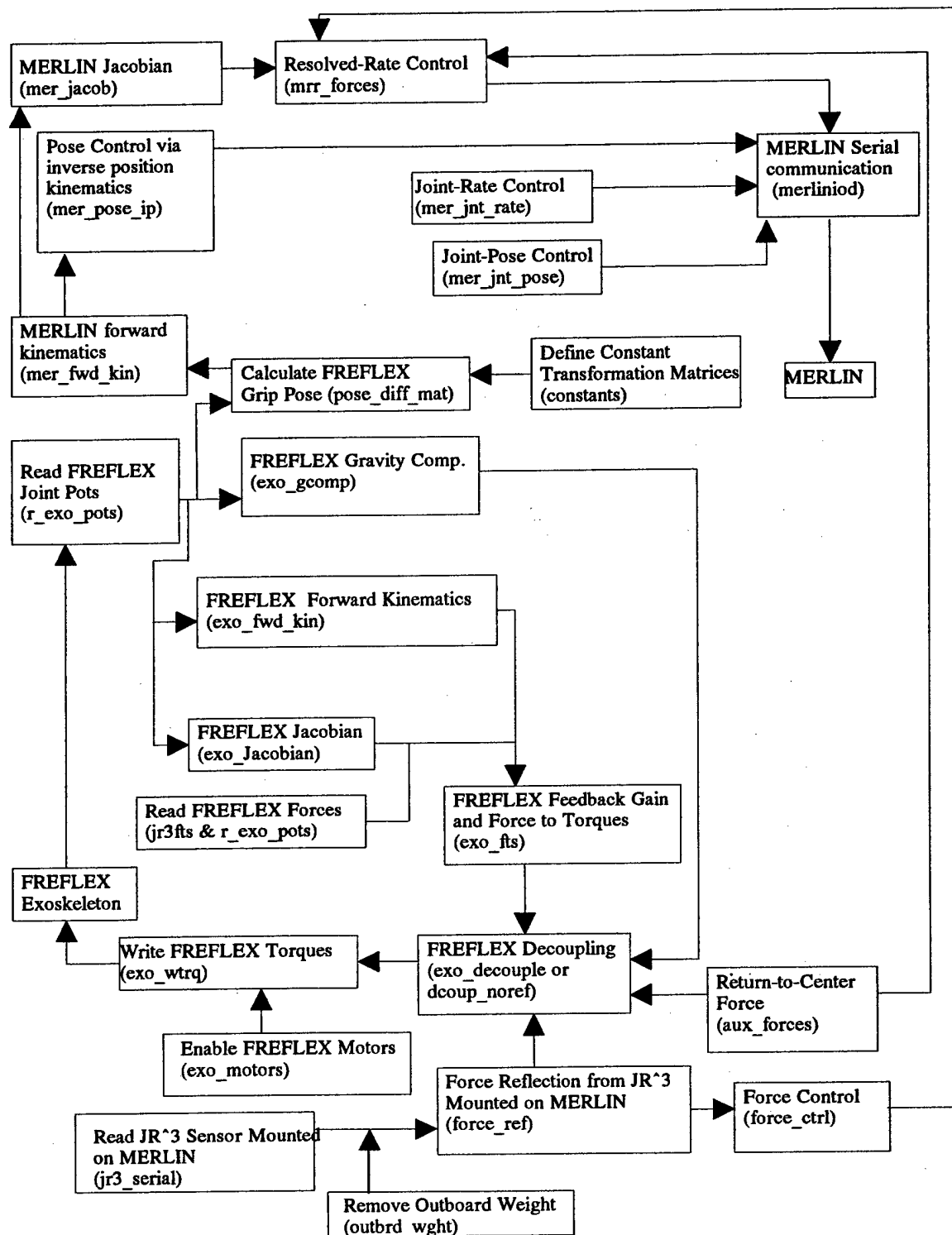
Figure 29. Teleoperation of the MERLIN Via the FREFLEX

Note that constants is turned-on when Chimera is activated and turns itself off after one cycle. This module defines the homogenous transformation matrices of the MERLIN which do not change. Constants can be used to command the MERLIN to its

home position at any time by turning-off the current control mode and then turning-on constants. Remember to depress the gray trigger on the FREFLEX to send the home position joint angles to the MERLIN. See Appendix A for more complete startup procedures.

The module outbrd_wght is also a one-time cycle. This cycle calculates the weight outboard of the JR^3 sensor mounted at the wrist of the MERLIN. This cycle should be initiated whenever JR^3 data is desired and while the MERLIN is in the home position. Note that exo_motors should be turned-off and the MERLIN foot pedal depressed while turning-on outbrd_wght.

The switches on the FREFLEX grip perform various tasks. The gray trigger and the black grip trigger are the only two switches that can be read on the FREFLEX grip by Chimera. These switches are read by the module exo_motors.

The black grip trigger and the FREFLEX foot pedal must be depressed to activate the FREFLEX motors for force reflection to the user.

The gray trigger has two uses. The first is a safety mechanism that only allows joint commands to be sent to the MERLIN if the switch is depressed. If the switch is depressed then merliniod cycles and sends the calculated joint angles to the MERLIN. This is the reason for turning off exo_motors to perform the outbrd_wght cycle and to operate under mer_jnt_pose. The modules outbrd_wght and mer_jnt_pose fool the MERLIN into believing that the gray switch is depressed and thus eliminate the need for the FREFLEX in these modes.

The gray trigger also serves as an indexing button in pose, joint rate, and resolved-rate control modes. The home grip position is redefined every time the gray trigger is depressed. The position at which the gray trigger is depressed becomes the new grip pose to which the users hand is led by the return-to-center forces during joint rate and resolved-rate control. This grip pose is also the new reference point for any FREFLEX displacements and rotations used to send joint commands to the MERLIN. The gray trigger allows the operator to access the entire workspace in pose control mode of the MERLIN even though the FREFLEX workspace is much smaller.

Joints of the MERLIN can be individually commanded to a desired position without using the FREFLEX. Turn-off other modules (mrr_forces, mer_jnt_rate, and mer_pose_ip) which can update MERLIN joint angles and turn-on mer_jnt_pose. The user is then prompted to input the joint (1-6, where 1 is the waist and 6 is the hand roll) to activate and the desired joint position in degrees. Again this requires that exo_motors be turned-off and the MERLIN foot pedal depressed while turning-on mer_jnt_pose.

The module mer_jnt_rate allows the user to command rates for any individual MERLIN joint via the FREFLEX. Turn-off other modules (mrr_forces, mer_jnt_pose, and mer_pose_ip) which can update the MERLIN joint angles and turn-on mer_jnt_rate. The user is prompted to enter the joint to actuate (1-6, where 1 is the waist and 6 is the hand roll). A vertical displacement from the grip home position defined by depressing the gray trigger on the FREFLEX grip corresponds to a joint rate for the desired MERLIN joint. For example if joint 1 is chosen for actuation, then an upward displacement of the FREFLEX grip will cause the MERLIN to rotate clockwise about its waist and a downward displacement of the FREFLEX grip will result in a counterclockwise rotation

of the waist. The MERLIN joint rate is proportional to the FREFLEX grip displacement and a return-to-center force helps the user locate the current neutral command position.

The module exo_decouple instead of dcoup_noref must be turned-on to reflect forces via the FREFLEX from the JR^3 sensor mounted on the MERLIN to the user. If no force reflection is desired and force_ref is cycling such as when operating in NTRFC mode, then turn on dcoup_noref instead of exo_decouple. The module force_ref should be started while the MERLIN foot pedal is depressed, the FREFLEX gravity compensation is activated, and only after jr3_serial and outbrd_wght have been turned-on. This allows any sensor noise due to the MERLIN and FREFLEX motors during normal operation to be accounted for as an offset.

CHAPTER 6 EXPERIMENTAL PROCEDURE AND RESULTS

This chapter discusses the teleoperation experiment used to compare resolved-rate control, the NTRFC control with force reflection, and the NTRFC without force reflection.

6.1 Experimental Design

The experimental objective and methods used to accomplish it are discussed in this section.

6.1.1 Objective

The objective of the experiment was to compare various modes of teleoperation implemented in the HSF Lab. The three evaluated modes were resolved-rate control and the NTRFC with and without force reflection. The parameters of comparison are time to complete the task and the work exerted by the manipulator on the environment as used by Williams and Aldridge (1997). This work term is the sum of all wrench components times the Cartesian displacement components during the task as shown in Equation 43:

$$W = \sum_{i=1}^{6} \left| \int_0^{\delta_i} F_{MRFi}(\xi_i) d\xi_i \right|. \qquad (43)$$

The work term provides a numerical value for the environmental contact wrench with consistent units for summation. Note that the work values are always positive since the MERLIN is performing the work on the taskboard.

The experimental design matrix is shown in Figure 30. A total of 120 tasks, 40 for each control mode, will be performed by each subject.

3 Control Modes
Evaluated
(Repeated Twice)

5 Trial Days

4 Task Difficulty Factors
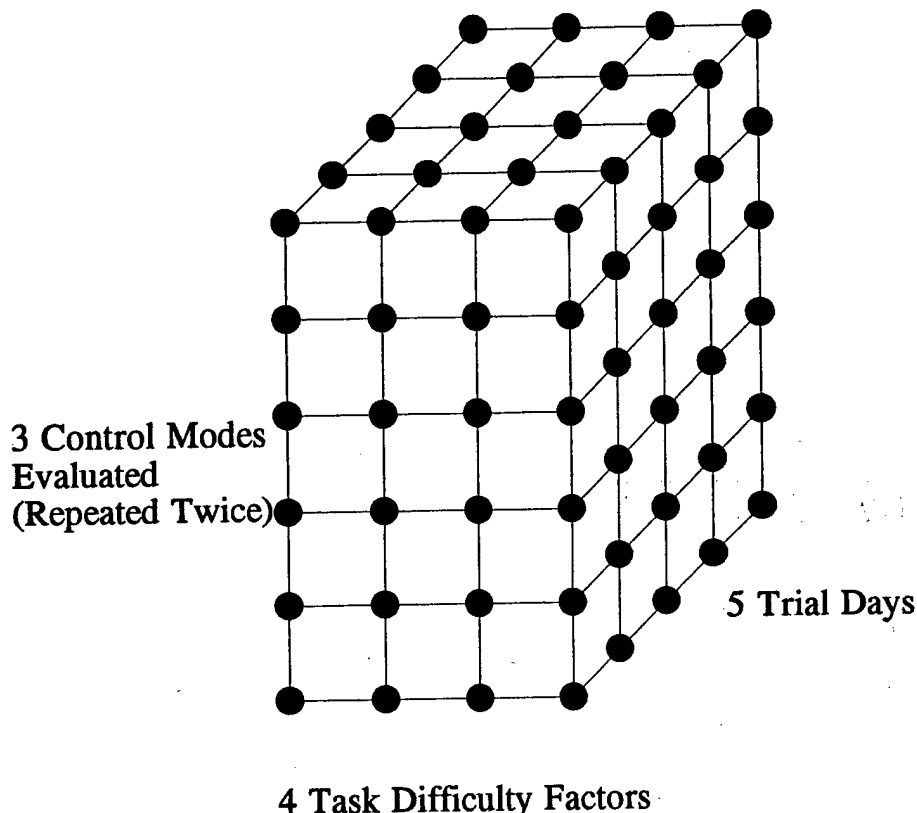
Figure 30. Experimental Design Matrix

6.1.2 Fitts' Law

Fitts' Law provides a method to classify the difficulty of the tasks being performed by the teleoperator. An index of task difficulty is found by Equation 44 (Fitts & Peterson, 1964):

$$I_D = \log_2 [(2 * A) / (D_h - D_p)] .$$
(44)

The variable A in Equation 44 is the amplitude of the task and the variables D_h and D_p are the hole and peg diameters respectively. Therefore the denominator formed by D_h and D_p is the clearance between the hole and the peg.

The standard unit of I_D is bits. Therefore when task time is plotted versus I_D, the inverse of the slope has units of bits per second which is also referred to as the baud rate. Baud rate is the capacity to perform a task. This task time versus I_D plot is linear within a certain range of task difficulty. If the task is extremely easy or difficult, the plot will be nonlinear (Dr. D. W. Repperger, personal communication, March 13, 1998). For example, threading a needle would result in a nonlinear trendline.

6.2 Experimental Setup

A general arrangement of the experimental equipment can be seen in Figure 31.

Figure 31. Experimental Equipment Arrangement
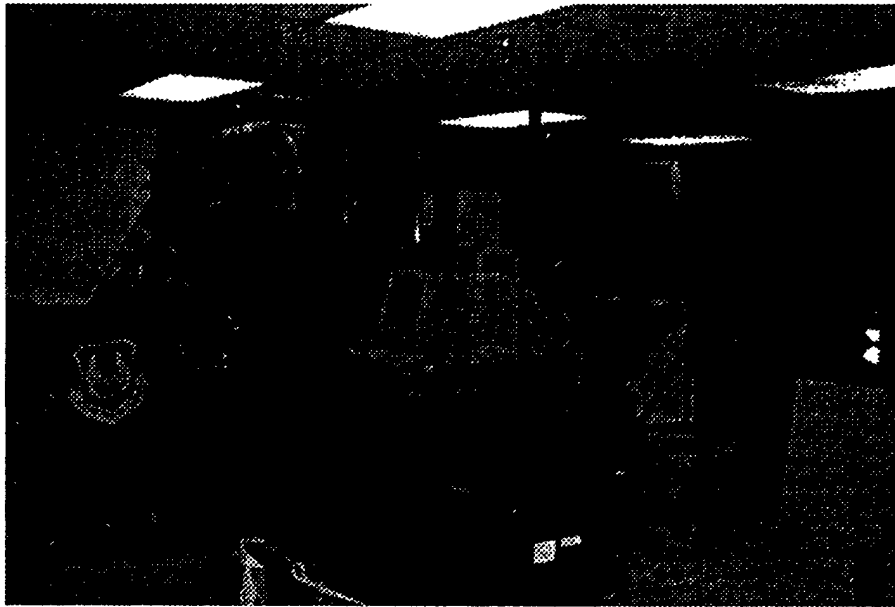
The operator is shown gripping the FREFLEX. The Sun SPARCstation which operates Chimera can be seen behind the operator's left arm and the rear of the television used to monitor the robot can be seen in front of the teleoperator. The MERLIN and taskboard are in the background on the right side of the photo. A closer view of the MERLIN and taskboard arrangement can be seen in Figure 32.

Figure 32. Taskboard and MERLIN Arrangement

The PC in the bottom left corner of Figure 32 controls the taskboard. The PC is indicating that the operator should move the peg into hole 2 and then proceed to hole 7. The orange and gray MERLIN control cabinet located in the corner can also be seen in the background of Figure 32.

6.2.1 Video

Two camera views of the taskboard provide visual feedback to the FREFLEX operator. The cameras available to the HSF Lab were too large for MERLIN wrist mounting. Therefore two cameras were used to supply visual telepresence to the FREFLEX operator. One camera was mounted from the ceiling above the MERLIN and the other was mounted to provide a side view of the taskboard. The operator can visually verify the peg pose with these two views.

Neither of these cameras have zoom capability and the side view camera yields a black and white image. Tape was placed on the floor to allow the taskboard to be positioned consistently throughout the experiments. This minimizes camera repositioning and provides the subjects with a consistent perspective.

A special-effects generator was used to divide the television display into two views. The side view of the taskboard was displayed on the top half and the over-the-shoulder view of the MERLIN was displayed on the bottom half of the television monitor.

Figure 33 is an example of the visual feedback that the teleoperator receives.
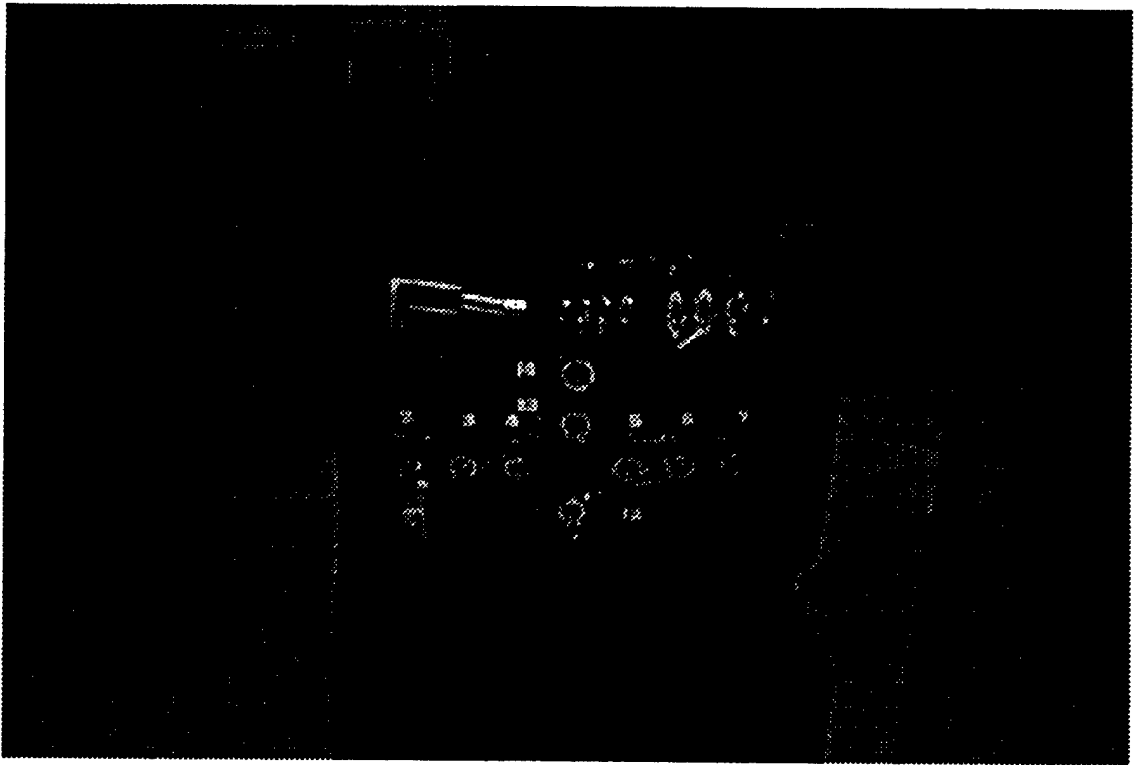
Figure 33. Teleoperator Visual Feedback

6.2.2 MERLIN Peg Holder

MERLIN uses a tool that was developed for previous HSF teleoperation experiments to hold the taskboard peg. The tool is shown in Figure 34.
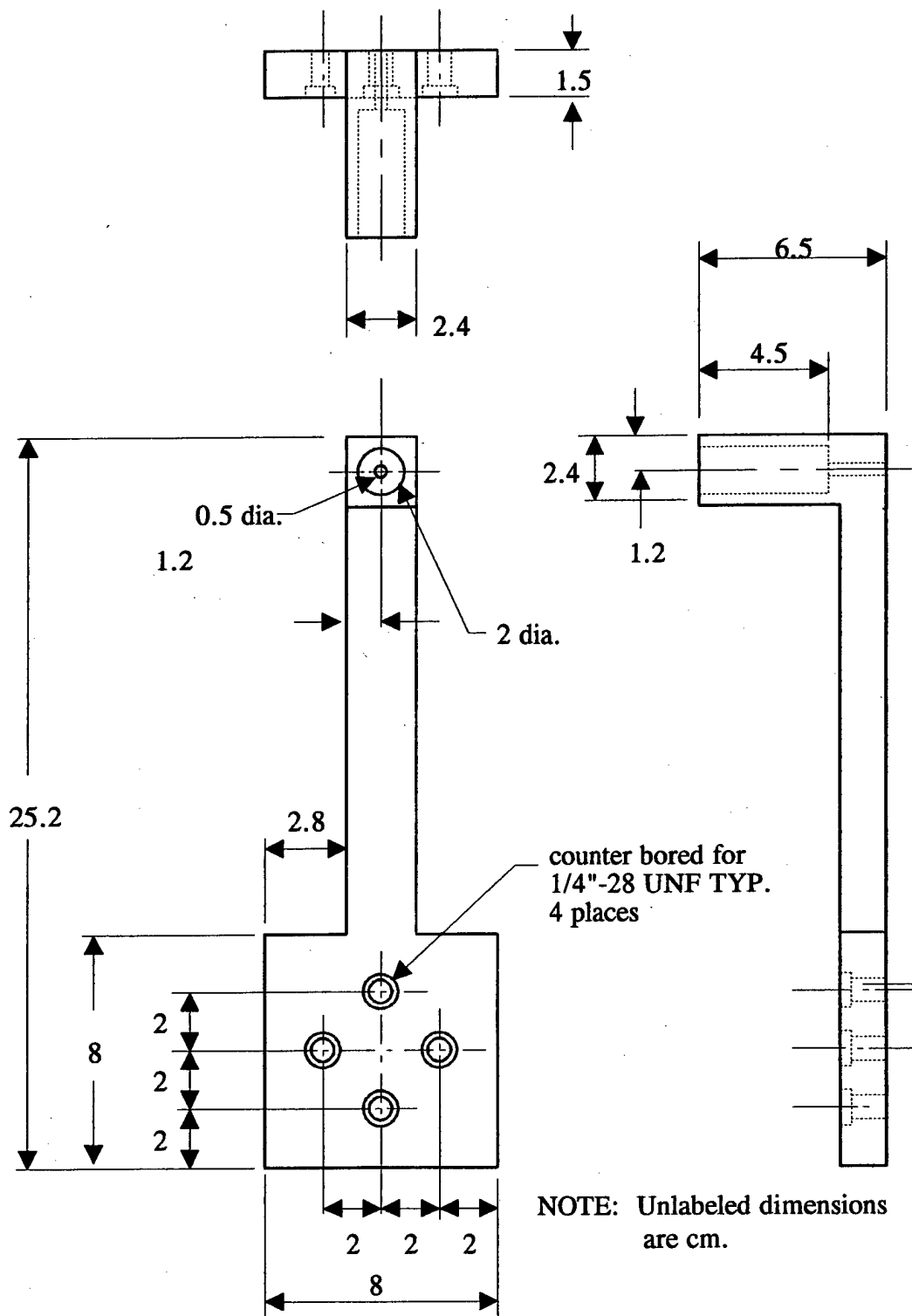
Figure 34. MERLIN Peg-Holding Tool

The L-shaped design of the tool allows an unobstructed over-the-shoulder view of the peg while operating the MERLIN in the elbow down position.

A relatively stiff spring has been inserted into the tool peg hole to decrease the chances of damaging the taskboard or robot by excessive contact forces. The peg is held in place by a bolt that passes through the 0.5 cm hole and screws into the rear of the peg. Since the bolt is 7.6 cm long, a 2.3 cm spacer between the back of the tool and the head of the bolt keeps the peg from falling out of the holder and still allows approximately 2.5 cm of spring compression. The spring also helps to compensate for system delays by compressing upon environmental contact which allows a split second for the robot or operator to react before the board will be physically moved by the MERLIN. The teleoperator receives an extra visual contact cue since the bolt that holds the peg inside the tool moves out the back of the tool due to the spring compressing when contact occurs.

6.2.3 Taskboard

The taskboard is operated by the 286 computer named "Achy". This computer reads the taskboard switches, indicates to the SPARCstation when the task has been started and ended, records the time required to complete each task, and prompts the operator with "beeps" when the taskboard switches have been activated. A high-pitch beep indicates that the switch in the task start hole has been activated and a low-pitch beep signals contact with the switch in the task destination hole. This PC can be seen in Figure 32.

The taskboard must have the same position for each trial to maintain consistent camera views. Tape has been placed on the floor to achieve this repeatable position.

Approximately 0.25 inch (0.635 cm) thick slices of 0.75 inch (1.905 cm) diameter wooden dowels were placed into holes 2, 6, and 7 to prevent the smaller peg from missing the switch once in the hole.

For taskboard operating procedures see Appendix A.

6.3 Experimental Procedure

The teleoperators in this experiment were instructed to complete the tasks as quickly as possible and with minimal contact with the taskboard. The operators were told when to enter the start hole and then to proceed to the destination hole. The 24 tasks completed each day were broken up into six groups of four tasks each. After each of these smaller task groups, the operator would get a brief break (usually two to five minutes) while control modes or pegs were changed. The time to complete a 24 task session ranged from 30 minutes up to 1 hour. The subjects were not informed which control mode was currently in operation and the order of the control modes varied from day to day.

The data from each subject's five sessions were evaluated for consistency. Usually the data of the first two sessions of a subject were considered practice and not included in the final results. However two of the seven subjects demonstrated the same competence for the final four sessions. Therefore four days of data were used for these subjects while the other five had three valid days of data.

6.4 Experimental Results

The results of the experiment to compare the teleoperation control methods installed in the HSF Lab are discussed in this section.

6.4.1 Task Time

Figure 35 displays the average time to complete the task of specified difficulty as determined by Equation 44. The results are shown for resolved-rate control without force reflection, the NTRFC without force reflection, and the NTRFC with force reflection. Each bar represents the average time of 46 tasks performed by seven research subjects.
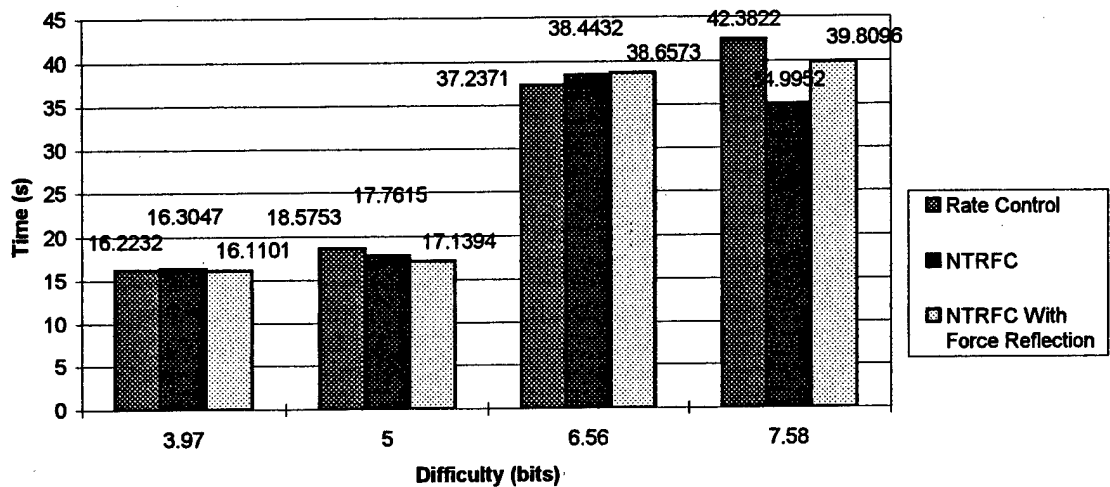
Figure 35. Average Task Time Versus Task Difficulty

Notice from Figure 35 that the control methods performed similarly until the most difficult task was performed. Only the most difficult task yielded significantly different results for the three control modes. The 90% confidence level for rate control performing the most difficult task is from 39.3653 seconds to 45.3991 seconds. The 90% confidence level for NTRFC without force reflection performing the most difficult task is from 33.4212 seconds to 36.5692 seconds.

Figure 36 verifies that this teleoperation experiment did result in a linear plot via Equation 44. The coefficients of determination range from 0.8155 to 0.9520.
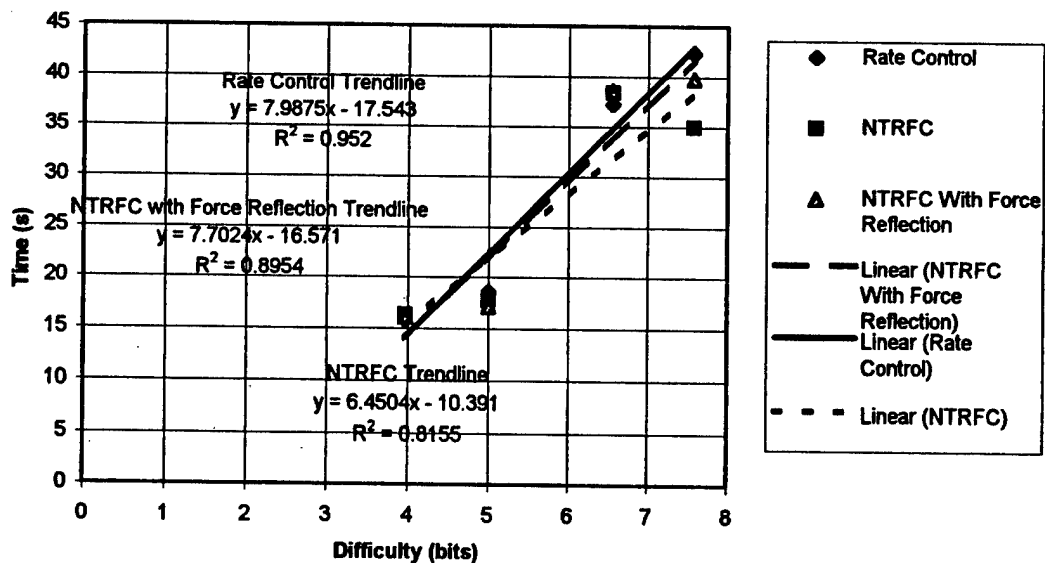
Figure 36. Average Task Time Versus Task Difficulty with Linear Trendlines

Figure 37 displays the sample standard deviations of the task time data. There is a pattern for the 3.97, 5, and 6.56 bit difficulty tasks. Rate control consistently has the lowest deviation while NTRFC with force reflection possesses the highest. The sample standard deviation of the task time data for NTRFC without force reflection was relatively low during the 7.58 bit task. The small sample standard deviation relative to the mean for the NTRFC during the most difficult task indicate that the performance is very consistent and repeatable.
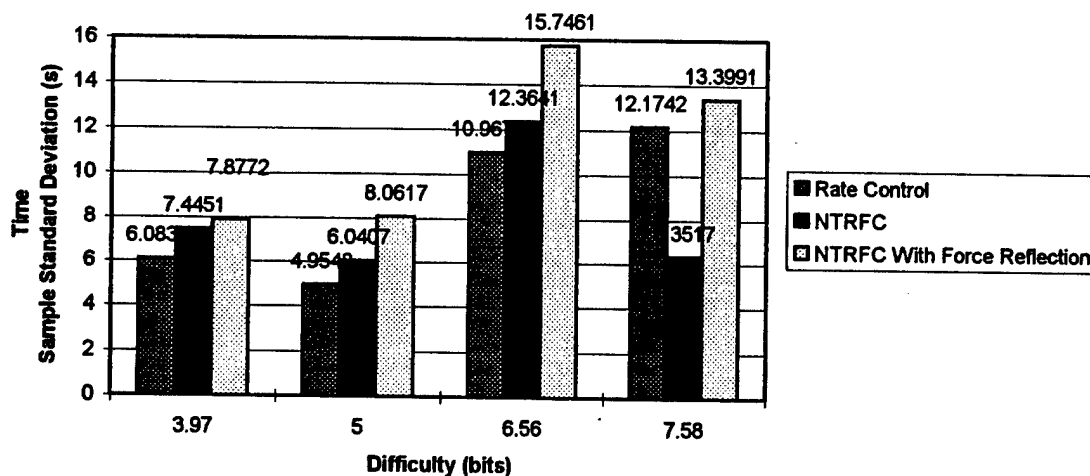
Figure 37. Task Time Sample Standard Deviation

6.4.2 Task Work

The average task work as calculated by Equation 43 is plotted in Figures 38 and 39 versus the task difficulty obtained from Equation 44. Each plotted value is the average of 92 tasks performed by seven subjects. The task work average increased linearly as the task difficulty increased as can be seen in Figure 39. Notice that the control method with the lowest average task work varied with task difficulty.
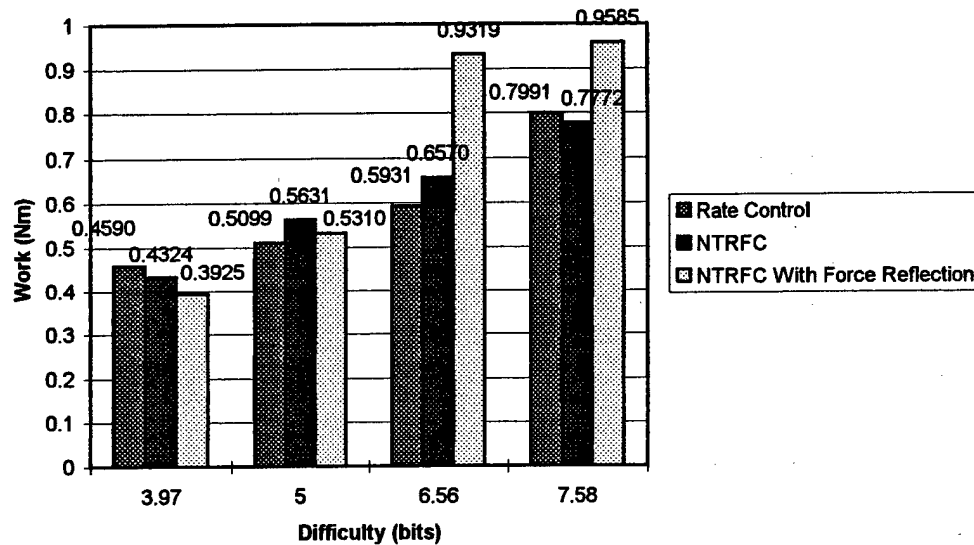
Figure 38. Average Task Work Versus Task Difficulty
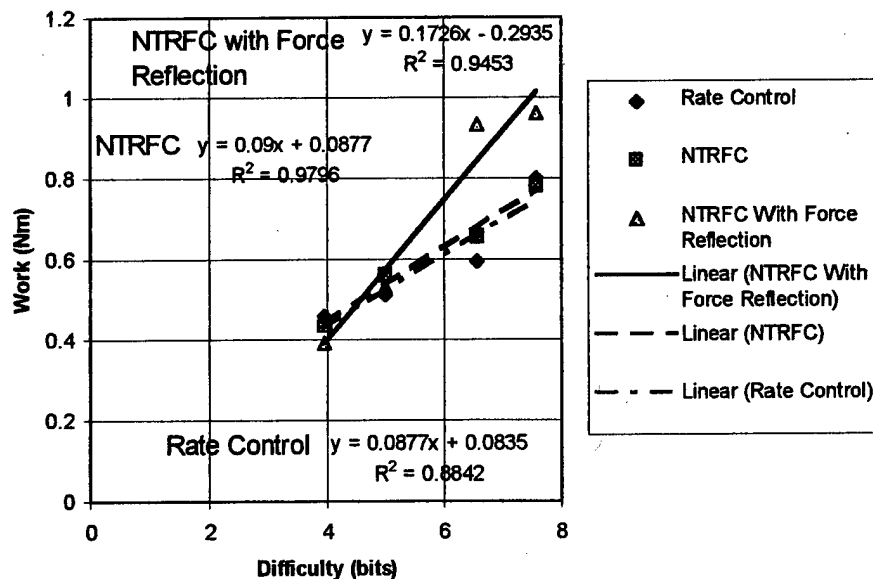
Figure 39. Average Task Work Versus Task Difficulty With Trendlines

67

The distance between the holes is less likely than the peg and hole clearance to affect the work values. Contact with the board usually results when trying to insert the peg and not when traversing between holes. The index of difficulty calculated by Equation 44 and used in Figures 38 and 39 is a function of the distance between the holes and the peg and hole tolerance. Figure 40 displays the average work as calculated by Equation 43 plotted versus peg and hole clearance. The graph shows that subjects usually completed the tasks with the small peg as expected with less work being exerted on the taskboard than with the large peg. The NTRFC with force reflection consistently resulted in the highest work values while the NTRFC without force reflection yielded the lowest values.
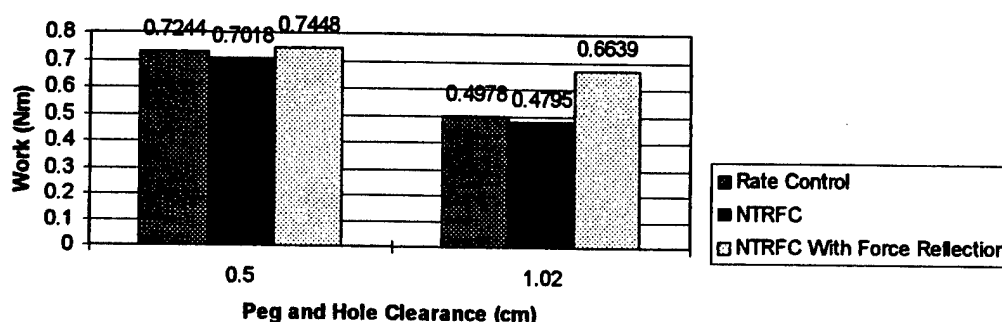
Figure 40. Average Work Versus Peg and Hole Clearance

The sample standard deviations for the work data are shown in Figures 41 and 42. Individual task work magnitudes that were 100 times the average work values for a particular operator sometimes were encountered. These values usually resulted from the subject ignoring one or more of the video or force-reflection feedback cues. This resulted in relatively large sample standard deviations when compared to the work average values from Figures 38, 39, and 40.
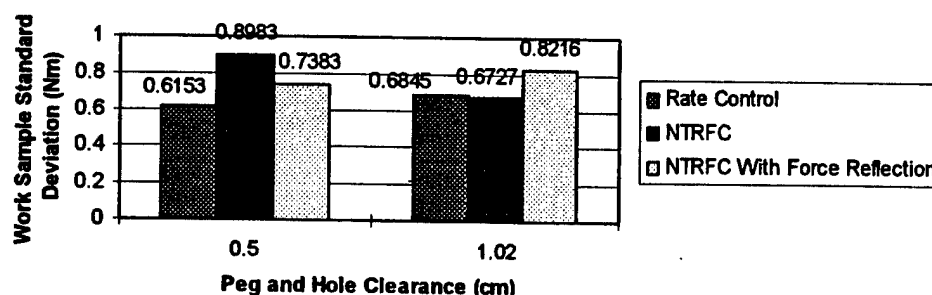
Figure 41. Work Sample Standard Deviation Versus Peg and Hole Clearance
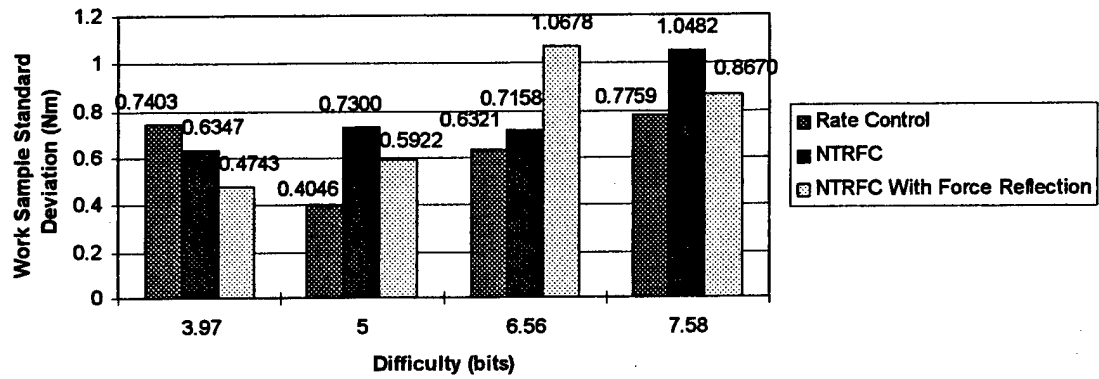
Figure 42. Work Sample Standard Deviation Versus Task Difficulty

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE

RESEARCH

This chapter summarizes the work presented, emphasizes significant conclusions, and recommends future research and system improvements.

7.1 Summary of Research

The top priority of the research discussed in this report was to advance the teleoperation research capabilities and knowledge of the HSF Laboratory at Wright-Patterson Air Force Base. The ability to control the MERLIN via the FREFLEX was implemented for the first time. The MERLIN can now be remotely controlled by joint rate control, joint position control, pose control, rate control, force control, and by the novel NTRFC. The FREFLEX with or without force reflection can control the MERLIN via joint rate control, pose control, rate control, and by the NTRFC.

Existing rate control methods are often preferable to inverse position control in free motion, but unacceptable in contact with the environment due to large contact forces. The NTRFC provides a method by which rate control in free motion naturally transitions to force control. The combination of the rate and FMA algorithms acting simultaneously on all Cartesian axes allows this capability. The NTRFC was modeled, simulated, and implemented during this research in the HSF Lab and preliminary evaluations have been made (See Chapter 6). The results of this research have provided the HSF Lab with another teleoperation tool and will allow them to further evaluate the NTRFC and other teleoperation control methods.

Data were collected for comparing rate control without force reflection and NTRFC with and without force reflection using a peg-in-hole taskboard. The experimental data can fulfill Research Thrust 4 of the U.S. Department of the Air Force protocol entitled "Force Reflection Studies in the HSF Laboratory" (1997).

7.2 Conclusions

1. The control architecture for force-reflecting teleoperation was successfully implemented using two devices in the HSF Lab which were never before interfaced. These devices, the FREFLEX and the MERLIN, can now perform the teleoperation tasks for which they were acquired. This installation has increased the ability of the HSF Lab to perform teleoperation experiments.
2. The teleoperation experimental data exemplify the need for a better method of choosing gains, both force-to-rate and force-reflection gains, for the NTRFC mode. The NTRFC with force reflection consistently performed poorly. The force-reflection gains were too large and caused the operator to overcompensate. For instance if the operator came into contact with the bottom of the hole, the FMA algorithm would command the robot to move the peg so that the MERLIN mounted JR^3 sensor was reading a force of zero

exerted on the peg from the bottom of the hole. However since the force-reflection gains were too large, the operator's hand was pushed upward by the FREFLEX force reflection causing the FREFLEX grip to move to a position that commands the robot to move the peg upward. Ideally the force reflection would return the grip to the deadband or not be strong enough to change the user's input but still alert the user of the contact. Theoretically if these gains were adjusted properly, the teleoperator's performance would be enhanced.

3. The NTRFC performance was adversely affected by a slow response of the MERLIN to the force-to-rate commands and of the FREFLEX to force-reflection commands from the MERLIN wrist-mounted JR^3 sensor. The MERLIN delay does not allow the FMA algorithm to respond to contact immediately thus increasing the task time and work. The FREFLEX does not provide immediate force feedback from the MERLIN sensor and therefore the operator loses this force reflection as a contact warning. This tends to increase the task work and time.

7.3 Future Research

The implementation of the force-reflecting teleoperation control architecture into the HSF Lab leads to unlimited future research opportunities.

1. Reducing the delay when sending commands from the MERLIN JR^3 to the FREFLEX and to the MERLIN during FMA is the top priority. Using a parallel connection instead of a serial connection between the sensor and the processing board Control, implementing a new JR^3 sensor that can reportedly transmit data at 8 kHz (JR^3, Inc., 1997), or replacing the current 33 MHz processing boards, Control and Crusher, with faster boards could reduce the delay.

2. The positive and negative moment reflection inconsistencies should be addressed to improve FREFLEX moment reflection. This would improve gravity compensation and the CFRTC force while operating in rate control mode.

3. Developing methods, perhaps adaptive control techniques, for proper NTRFC gain determination is important to reduce the dependence on models and heuristic adjustments. Analytical techniques for NTRFC design in nonlinear systems should also be investigated. These developments would make the NTRFC easier to implement for specific teleoperation tasks.

4. One interesting research project could use the JR^3 present on the wrist of the FREFLEX to compare the force exerted by the teleoperator on a wall with the force exerted by the MERLIN on another wall in NTRFC mode. In theory the forces would be directly proportional.

5. Subjects in the HSF Lab found pose control initially more intuitive than rate control. Another project could apply the FMA algorithm to a pose control mode instead of a rate control mode. This would allow the teleoperator to operate in pose control mode until the moment of contact and then the robot

would be given additional force-accommodation commands in rate mode as in the NTRFC. This could easily be implemented by adjusting existing gains and activating the correct Chimera modules.

6. Parabolic rate gains would be another improvement to the current system. Using parabolic gains during rate control could allow the teleoperator to move rapidly while in free space and also make slow and precise movements when needed. Currently the gains are linear and must be relatively low to allow for the peg to be inserted in the taskboard. This limits the maximum speed that can be commanded. However if the linear gains are increased, the peg can be moved quickly but it is difficult to align the peg for insertion.

7. The rate control mode could be improved with the addition of SVD to find a solution when the robot is in a singular configuration. Currently the teleoperator must avoid the singularities while performing a task.

8. The teleoperation experiment discussed in Chapter 6 revealed several notable areas of consideration for future experimentation. The experiments evaluated general control methods that were not designed specifically to perform the peg-insertion task. The performance for the peg-insertion task could be improved by adding virtual walls for pure X, Y, and Z motion. Minimizing the possible Cartesian and wrist rotation commands to the maximum number required to perform the task could also improve teleoperator performance. For instance the wrist roll command is not needed for single peg insertion.

9. The spring in the peg holder adds compliance to the system. There is a debate whether or not this compliance causes the data to appear similar for the various control modes. The experiment could be repeated without the spring in the peg holder to resolve this issue.

10. Quality visual feedback is essential in teleoperation especially if force reflection or force accommodation is not activated. The experimental visual display was not ideal. A small camera mounted on the robot near the peg and a broad angle side view would probably result in a better teleoperation perspective.

11. In future teleoperation experiments, audio feedback from the MERLIN could be eliminated. The experimental setup allowed the operator to hear the MERLIN operating and contacting the taskboard.

12. Changing the CRF could also affect teleoperation performance. The experiments discussed in this report involved the teleoperator controlling the robot in the MRF frame which was located at the tip of the peg. Some subjects had difficulty visualizing this frame. Another experiment could be performed with the CRF being a frame located on the taskboard which is stationary.

REFERENCES

American Robot Corporation. (1985, February). <u>Service manual for the System II Merlin® intelligent robot.</u> (Manual No. MSII-SER). Pittsburgh, PA: Author.

American Robot Corporation. (1986, August). <u>High speed host interface, reference manual.</u> (Issue 1). Pittsburgh, PA: Author.

American Robot Corporation. (1995a). <u>AARM Motion™ industrial robot controller, teach pendant user manual.</u> (Issue 1.2.). Oakdale, PA: Author.

American Robot Corporation. (1995b). <u>AARM Motion™ retrofit for the Merlin® industrial robot MUCII 64 controller, user and service manual.</u> (Issue 1.1). Oakdale, PA: Author.

American Robot Corporation. (1996, September). <u>AR-Basic™ V6.0 for the AARM Motion™ industrial robot controller, programming manual.</u> (Issue 1.7). Oakdale, PA: Author.

American Robot Corporation. (1997, July). <u>HSHI high speed host interface, reference manual.</u> (Issue 2.0). Oakdale, PA: Author.

Bryfogle, M.D. (1990). <u>Force reflection algorithms for exoskeleton controllers</u> (US Air Force Technical Report AAMRL-TR-90-090). Wright-Patterson AFB, OH.

Colbaugh, R., Seraji, H., & Glass, K. (1993). Direct adaptive impedance control of robot manipulators. <u>Journal of Robotic Systems, 10</u> (2), 217-248.

Committee on Advanced Robotics for Air Force Operations, Air Force Studies Board, Commission on Engineering and Technical Systems, & National Research Council. (1989). <u>Advanced robotics for air force operations</u> (Contract No. F49620-87-C-0122). Washington, DC: National Academy Press.

Craig, J.J. (1989). <u>Introduction to robotics: Mechanics and control.</u> Reading, MA: Addison Wesley Pub. Co.

Fitts, P.M. & Peterson, J.R. (1964). Information capacity of discrete motor responses. <u>Journal of Experimental Psychology, 67</u> (2), 103-112.

Goldenberg, A.A., Apkarian, J.A., & Smith, H.W. (1989). Approach to adaptive control of robot manipulators using the computed torque technique. <u>ASME Journal of Dynamic Systems, Measurement and Control, 111</u> (1), 1-8.

Hogan, N. (1985, March). Impedance control: An approach to manipulation. ASME Journal of Dynamic Systems, Measurement, and Control, 107, 1-24.

Huang, M.Z. (1993). Efficient coordination of an anthropomorphic telemanipulation system (US Air Force Technical Report AL/CF-TR-1995-0120). Wright-Patterson AFB, OH.

Hyde, J.M. & Cutkosky, M.R. (1994). Controlling contact transition. IEEE Control Systems Magazine, 14 (1), 25-30.

Industrial Drives. (1993). BDS4 series, installation and setup manual. (M93100 - Issue 2). Radford, VA: Author.

Ingimarson, D., Stewart, D.B., & Khosla, P.K. (1995, February). Chimera 3.2 the real-time operating system for reconfigurable sensor-based control systems. Pittsburgh, Pa: The Robotics Institute, Dept. of Electrical and Computer Engineering at Carnegie Mellon University.

Johnsen, E.G. & Corliss, W.R. (1971). Human factors applications in teleoperator design and operation. New York: John Wiley & Sons.

JR^3, Inc. (1988, May). Universal force-moment sensor system operation manual. (Rev. 2.0a). Woodland, CA: Author.

JR^3, Inc. (1997). Multi-Axis Load Cell Technologies [Internet]. Available: WWW.JR3.COM

Kane, T.R., Likins, P.W., & Levinson, D.A. (1983). Spacecraft dynamics. New York: McGraw-Hill.

Maciejewski, A.A. & Klein, C.A. (1989). The Singular Value Decomposition: Computation and Applications to Robotics. The International Journal of Robotics Research, 8 (6), 63-79.

Odetics, Inc. (1991, April). Exoskeleton master arm, wrist, and end effector controller with force-reflecting telepresence. (Small business Innovative Research Program, Contract No. F33615-89-C-0587, Phase II Interim Report). Anaheim, CA: Author.

Paul, R.P. (1981). Robot manipulators. Cambridge, MA: MIT Press.

Raibert, M. & Craig, J.J. (1981, June). Hybrid position/force control of manipulators. ASME Journal of Dynamic Systems, Measurement, and Control, 102 (2), 126-133.

Repperger, D.W. (1991, January). Active force reflection devices in teleoperation. IEEE Control Systems Magazine, 11, 52-56.

Repperger, D.W. (1995). Biodynamic and spasticity reduction in joystick control via force reflection (US Air Force Technical Report AL/CF-TR-1995-0152). Wright-Patterson AFB, OH.

Repperger, D.W., Phillips, C.A., & Chelette, T.L. (1995). A study on spatially induced "virtual force" with an information theoretic investigation of human performance. IEEE Transactions on Systems, Man, and Cybernetics, 25 (10), 1392-1404.

Repperger, D.W., Phillips, C.A., Hill, B.O., & Roark, M.R. (1996). Design of haptic interfaces via a power ellipsoid approach (Wright-Patterson AFB Human Sensory Feedback Lab Internal Report).

Repperger, D.W., Scarborough, E.L., & Chelette, T.L. (1991). Construction of a dual axis force reflection stick and test station (US Air Force Technical Report AL-TR-1992-0041). Wright-Patterson AFB, OH.

Rosenberg, L.B. (1992). The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments (US Air Force Technical Report AL/CF-TR-1994-0089). Wright-Patterson AFB, OH.

Rosenberg, L.B. (1993). The use of virtual fixtures to enhance operator performance in time delayed teleoperation (US Air Force Technical Report AL/CF-TR-1994-0139). Wright-Patterson AFB, OH.

Skaar, S.B., & Ruoff, C.F. (Eds.). (1994). Teleoperation and robotics in space. Washington, DC: American Institute of Aeronautics and Astronautics.

Spain, E.H. (1989, March). Peg-in-Hole taskboard documentation. (Version 1.0). Unpublished manuscript.

Tarn, T.J., Wu, Y., Xi, N., & Isidori, A. (1996). Force regulation and contact transition control. IEEE Control Systems Magazine, 16 (1), 32-40.

Tzafestas, S.G. (Ed.). (1991). Intelligent robotic systems. New York: Marcel Dekker.

U.S. Department of the Air Force Armstrong Laboratory. (1997). Force Reflection Studies in the HSF Laboratory (HURC protocol number 95-10). Wright-Patterson Air Force Base, OH: Author.

Vertut, J. & Coiffet, P. (1986). Teleoperation and robotics applications and technology (Vol. 3B) (I. Aleksander, Trans.). Englewood Cliffs, NJ: Prentice-Hall. (Original work published 1985).

Vukobratovic, M. & Stojic, R. (1996). On position/force control of robot interacting with dynamic environment in Cartesian space. ASME Journal of Dynamic Systems, Measurement and Control, 118 (1), 187-92.

Whitney, D.E. (1969, March). Resolved motion rate control of manipulators and human prostheses. IEEE Transactions on Man-Machine Systems, MMS-10 (1), 47-52.

Whitney, D.E. (1985). Historical perspective and state of the art in robot force control. IEEE International Conference on Automation and Robotics (pp. 262-268). Silver Spring, MD: IEEE Computer Society.

Williams, R.L., II & Aldridge, H.A. (1997). Cartesian control of force-reflecting hand controllers. Conference on Applied Mechanisms and Robotics, Cincinnati, OH.

Williams, R.L., II, Harrison, F.W., & Soloway, D.I. (1996). Naturally-Transitioning rate-to-force controller for manipulators. 1996 IEEE International Conference on Automation and Robotics, Minneapolis, MN.

Williams, R.L., II, Harrison, F.W., & Soloway, D.I. (1997). Shared control of multiple-manipulator, sensor-based telerobotic systems. IEEE International Conference on Automation and Robotics, Albuquerque, NM.

Willshire, K., Harrison, F.W., Hogge, E.F., Williams, R.L., II, & Soloway, D.I. (1992) Results of telerobotic hand controller study using force information and rate control (AIAA Paper 92-1451). AIAA Space Programs Conference, Huntsville, AL.

Yao, B. & Tomizuka, M. (1995). Adaptive control of robot manipulators in constrained motion-controller design. ASME Journal of Dynamic Systems, Measurement and Control, 117 (3), 320-328.

APPENDIX A

OPERATING PROCEDURES

Appendix A

Operating Procedures

Various operating procedures for the FREFLEX and MERLIN system are presented in this Appendix.

A.1 FREFLEX Operating Procedures

The FREFLEX operating procedures and troubleshooting methods are discussed in Section A.1.

A.1.1 FREFLEX Hardware Power Up

The following steps illustrate the power up procedure for the FREFLEX exoskeleton:

1. Turn on the circuit breaker in the power panel labeled
   PPL
   120-208V 3@ 4W
   FM PDP-A.
   The circuit breaker is labeled FREFLEX at positions 8, 10, and 12.
2. Power up the VME chassis at the rear of the cabinet.
   Note: Failure to connect the digital to analog cable to the VMIVME board 4100 has resulted in FREFLEX damage in the past.
3. Power up the FREFLEX interface chassis at the front of the unit which is located above the VME chassis.
   Note: The emergency stop button should be out and if the fault led is lit then the reset button should be depressed.
4. Power up the FREFLEX base unit by activating the circuit breaker that is located to the right of the power cord on the base.
   Note: Press the yellow reset switch on the base if lit for operation. For activation, the foot and the grip switches must be depressed and the software controlled safety relay must be running. The signal connectors below the FREFLEX grip should be checked before operation since these are easily disconnected and prevent the activation of the exoskeleton.
5. Undo steps 4 through 1 to shutdown the system.

A.1.2 FREFLEX Software Start Up

The software for operating the FREFLEX in Chimera 3.2 can be started using the following procedure:

1. Login under the Chimera account on the Sun SPARCstation.
2. Change to the freflex_merlin binary directory by typing "cd chim_3.2/freflex_merlin/bin" at an xterm window. Note that the home

directory is /usr/chimera. Start Chimera 3.2 by typing "chim" and execute the batch file to download both processors by typing "<exo" at the chim prompt.

3. Start the enet socket window for data display by typing "cd/usr/chimera/chim_3.2/freflex_merlin/src/enet" from another xterm window. Activate the data display program by entering "enet_disp" in this window. Note: The enet display needs to be started before RTPU Control is activated since Control spawns the sunenet routine which tries to connect to enet_disp. Start enet_data instead of enet_disp to collect and save data to the Sun disk. This step must be completed before the sundata routine is started and after loading the freflex_merlin program in step 2 since the sundata routine tries to connect to enet_data. Finally start the data transfer routine, sundata, by typing "on sundata" at the chim prompt.

4. Start execution on Control by typing "go control" at the chim prompt. Type "status" to see which programs are running and type "on *filename*" and "off *filename*" to turn programs on and off respectively.

5. Type "quit" at the chim prompt to terminate the programs and disable the FREFLEX. Typing "quit" again will terminate Chimera 3.2.

A.1.3 FREFLEX Troubleshooting

The qs server may remain running if the system ends abnormally. The program chimclean can be used to clean-up the system and terminate the qs server by typing "chimclean" at the operating system prompt.

Sometimes after rebooting the system a message appears when Chimera 3.2 is started that says "sxm drivers are not installed". If this occurs or the system locks up while trying to start Chimera 3.2, then the sxm drivers need to be installed using the following procedure:

1. Change to the Sun4c driver directory in an xterm window by typing "cd /usr/chimera/chim_3.2/sun4c/src/driver".
2. Login as the superuser.
3. Type "sxm.INSTALL" at the superuser prompt, %, to execute the batch file.
4. If no error messages are displayed and the display indicates that the sxm driver has been loaded, then type "exit" to logout as superuser.
5. Restart the Chimera 3.2 software.

If the system operates but the data output is drifting or not acting as expected, the FREFLEX pots should be checked for proper calibration and operation. While the system is running, place the FREFLEX in a fixed position. Select the pots enet display screen by typing "2" on the enet display. If any of the angles are drifting, the pot is not working properly. Use a voltmeter on the pot to identify the problem.

To check the pot calibration, rotate each joint until the calibration marks on the joint line-up. Joints one, three, and five should read zero degrees, joints two and six should read 90 degrees, and joint four should read -90 degrees when at their respective calibration positions. If a pot needs recalibrated, record the hexadecimal A/D number displayed for the corresponding pot on the enet pots display screen by typing "2" on the

enet display. Then open the file jointpot.sai in the directory /usr/chimera/chim_3.2/freflex_merlin/config and change the cal_cnts number for the corresponding pot to the recorded hexadecimal number.

Sometimes the FREFLEX will pulse when sending commands to the MERLIN. This requires a restart of the FREFLEX and MERLIN system. See Section A.3.

If the FREFLEX motors are activated by depressing the foot pedal and gray grip switch while the FREFLEX grip is in the volume defined as the human workspace, the red output fault lights on the FREFLEX motor drive boards will turn-on. The switch on the side of the FREFLEX must be turned-off and then back on to resume operation.

A.2 MERLIN Operating Procedures

The MERLIN operating procedures and troubleshooting methods are discussed in Section A.2.

A.2.1 Powering Up The MERLIN

The procedure for powering up the MERLIN is listed below:

1. Turn on the circuit breaker, switch 21, in the power panel on the east wall labeled

<div align="center">

PPL

120-208V 3@ 4W

FM PDP-A.

</div>

2. Ensure that the CRT and keyboard are connected to the PC inside the MERLIN control cabinet and that the CRT is powered up.
3. Turn on the circuit breaker at the bottom right-hand side on the front of the MERLIN control cabinet.
4. Pull out the emergency power button on the operator's panel on the front of the cabinet.
5. Set the key at the bottom of the operator's panel to *run*.
6. Depress the *POWER ON* button on the operator's panel.
7. Depress the *MOTOR POWER* button on the operator's panel.
   Note: The PC controller should now boot up in MS-DOS mode within a few seconds.
8. At the C:> prompt, type "I" to load the second processor and to start the server. Type "A" to start AR-Basic.
   Note: If Ctrl-Alt-Del is used to reset the computer, the second processor does not need to be reloaded.
9. Type "robot on" to set the robot to online.
10. Type "calibrate init" at the command prompt within AR-Basic. While the light fence is activated and the foot pedal is depressed, use the teach pendant to move the waist to the middle of the desired operating range and position the arm and wrist straight-out. To move the MERLIN arm using the pendant, the button *jnt* must be pressed and then the button *pos* or *ori* must be held while

using the three control bars on the right side of the pendant to actuate the desired joint. Press the *F1* function key on the keyboard while maintaining the light fence active and the foot pedal depressed. When the command prompt reappears, the calibration is complete.

11. To power off, type "exit" in AR-Basic and undo step 3 and then step 1.

A.2.2 Merlin Troubleshooting

If the wrist joints are emitting "grinding" sounds while moving and not moving smoothly, check that the robot is receiving proper plant air pressure. The air pressure might also be low if a hissing sound is heard when the MERLIN foot pedal is depressed.

If the robot encounters a limit switch during operation, it can go off-line. To alleviate this problem turn the key on the operator's panel to "BR OV" to override the brake and then manually rotate the robot into its workspace. Return the key to "RUN" to operate the MERLIN.

The MERLIN will not respond if the commanded position requires large joint motions from the current MERLIN position. For example if the MERLIN foot pedal is not depressed while the FREFLEX is sending commands to the MERLIN, the MERLIN will often not respond when its foot pedal is depressed. When this happens the MERLIN will be louder than normal when the pedal is depressed. If the MERLIN is louder than normal but slowly moving, send small commands to the wrist and the MERLIN will usually catch-up with the commands and then operate normally. The entire system requires restarting if the MERLIN does not recover.

A.3 FREFLEX Commanding MERLIN

The procedure for controlling the MERLIN manipulator via the FREFLEX is listed below:

1. Follow steps 1 through 10 for powering up the MERLIN.
2. Exit from AR-Basic by typing "exit" and type "hshicom" at the C:>AR-Basic> prompt.
3. Follow steps 1 through 4 for powering up the FREFLEX hardware.
4. Follow steps 1 through 4 for starting the FREFLEX software. Depress the MERLIN foot pedal and turn on the driver by typing "on merliniod".
5. Depress the MERLIN foot pedal and the gray trigger on the FREFLEX to send the MERLIN to its home position.
6. Activate the desired control mode by turning on the necessary modules. See Section 5.2.2 and Figure 28 for more information.
7. To shutdown the system via the serial link from the controlling processor, type "exit" at the chim prompt. If HSHI locks up, Ctrl-Alt-Del will reboot MS-DOS. Reverse step 3 and then step 1 for powering up the MERLIN. Follow step 5 from Section A.1.2 and then step 5 from Section A.1.1 to shutdown the FREFLEX.

A.4 MERLIN Mounted JR³ Sensor

The outboard weight and the noise due to the FREFLEX motors must be removed before the sensor data is reflected to the teleoperator or used to control the MERLIN. The procedure for preparing the sensor is listed below:

1. Turn on the sensor by activating the switch on the side of the JR³ electronics power enclosure. Note that the sensor should be turned off and then back on every time Chimera has been restarted.
2. Follow the directions for starting the FREFLEX and MERLIN as described in Section A.3 but stop after completing step 5.
3. After the MERLIN is in the home position, turn on the module to read the JR³ via the serial line by typing "on jr3_serial". Verify that the sensor is being read by choosing screen 8 of the enet display.
4. Turn off exo_motors by typing "off exo_motors". This will allow commands to be sent to the MERLIN without the FREFLEX gray trigger being depressed.
5. Depress the MERLIN foot pedal and turn on the module to remove the outboard weight from the sensor readings by typing "on outbrd_wght". After the message "Off outbrd_wght" is displayed, the MERLIN foot pedal can be released.
6. Turn exo_motors back on by typing "on exo_motors".
7. Depress the MERLIN and FREFLEX foot pedals and the FREFLEX black grip switch so that gravity compensation is activated on the FREFLEX. While activating these switches, turn on the module that removes the sensor offset and the noise due to the FREFLEX motors by typing "on force_ref".

The sensor may now be used to control the MERLIN via force control or reflect forces and moments to the teleoperator via the FREFLEX.

Type "on exo_decouple" and "off dcoup_noref" to reflect forces and moments from the MERLIN mounted sensor to the teleoperator via the FREFLEX.

Activate the modules as shown in Figure 28 to operate in force control mode. The rate control modules as well as the module force_ctrl must be activated for force control.

A.5 Taskboard

The steps required to read the taskboard are listed below:

1. Turn on the 286 PC labeled Achy.
2. Synchronize time of the taskboard PC to that of the Sun SPARCstation.
3. At the C:\WOS prompt type "taskbrd" to begin the taskboard program.
4. Enter the name of the configuration file which contains the experimental task plan. Note that the file used in the experimentation for this thesis is named murphy.1.
5. The data file name is created from the subject's initials, session number, and Julian date. JEB082040.raw is a typical data file name.

6. Type "on work" to record the work performed during a task. The module force_ref must be operating for the work to be measured. Follow the JR^3 procedure above to activate force_ref.

7. Skipped trials are not written on the 286. However if the work values are being recorded by the Sun, then the work values of these skipped trials are written to the file Taskboard.dat. Therefore step 1 is crucial. The task time data from the 286 PC and the work data from the Sun can be matched. Both the Sun and 286 data files are opened, appended, and then closed at the end of each trial.

APPENDIX B

CHIMERA MODULES

TABLE B-1. Chimera Modules

Module	In variables	Out variables	Description
aux_forces.c	Mer_cmd_mat, Rot_vel_mer, Dm_cycle, Merlin_enab	Mer_cmd_mat, Ret_to ctr, Damp_exo, Rot_vel_mer, Vel_mer, Dm_cycle	Calculates the return-to-center forces and moments and attempts to calculate the damping forces and moments. Is also necessary for rate operation since the linear velocities and rotational velocities of the mrf in mrf coordinates (Rot_vel_mer and Vel_mer) are calculated.
constants.c	Tm6, Th6, Tmh	Tcwo, Tbwo, T0b, Th6, Tmh, Tm6, Qm6, Ts6, Alpha, Theta, A, D	Generates the constant transformation matrices.
dcoup_noref.c	Gtrq_exo, Etrq_exo, Rmat_exo, Qvec_exo, Ret_to_ctr, Jacob_exo	Dtrq_exo	Decouples FREFLEX joint torques but does not allow wrench reflection from the MERLIN mounted JR^3 sensor to the FREFLEX operator.
enet_disp.c			Displays information in a Sun window.
exo_Jacobian.c	Qvec_exo, Zmat_exo	Jacob_exo	Calculates the FREFLEX Jacobian.
exo_ctrq.c	none	Ctrq_exo	Enable closed loop torques.
exo_decouple.c	Gtrq_exo, Etrq_exo, Rmat_exo, Qvec_exo, Ret_to_ctr, Damp_exo, Jacob_exo, Fref_wrtm	Dtrq_exo	Decouples FREFLEX joint torques including wrench from the MERLIN mounted JR^3 sensor.

Table B-1. (cont.)

Module	In Variables	Out Variables	Description
exo_forces.c	Fm_Mwrist, Tm_Mwrist, Fm_Melbow, Qvec_exo, Rmat_exo	Elbow_exo, Grip_exo	Rotates FREFLEX elbow and wrist forces and moments measured in the sensor frames to frame {7} & then transforms to origin of frame {0} in which the Jacobian matrix is described.
exo_fts.c	Jacob_exo, Elbow_exo, Grip_exo, Elbow_dsrd, Grip_dsrd, EXO_ELBOW, EXO_TORQ	Err_trq_exo	Calculates the joint torques due to the FREFLEX grip force errors measured from the JR^3 sensor at the FREFLEX grip. Also calculates the joint torques due to the elbow force error measured from the FREFLEX elbow sensor. These torques are then superimposed for each joint.
exo_fwd_kin.c	Am_Mjoints	Qvec_exo, Rmat_exo, Zmat_exo, ExoFWD_STOP	Calculates kinematic arrays from the seven FREFLEX joint angles.
exo_gcomp.c	Qvec_exo, Rmat_exo, Zmat_exo	Gtrq_exo	Calculates FREFLEX gravity compensation torques.
exo_halfg.c	Qvec_exo, Rmat_exo, Zmat_exo	Gtrq_exo	Calculates half of the required FREFLEX gravity compensation torques. This module was used in studies where the FREFLEX operator became too fatigued to perform the tasks using the FREFLEX with no gravity compensation. It requires the operator to support half of the FREFLEX weight.

Table B-1. (cont.)

Module	In Variables	Out Variables	Description
exo_motors.c	none	WOS_RECORD, Merlin_enab	Enables the FREFLEX motors through the Digital I/O card, VMIVME 2510B. This module also reads the FREFLEX grip switches and foot pedal. Dynamic braking occurs when the black grip switch or the foot pedal is released.
exo_wtrq.c	exo_Dtrq, exo_trq_max	exo_trq_max	Writes torques to the FREFLEX motors through the VMIC 4100 digital to analog converter board.
force_ctrl.c	Fref_wrtm, Merlin_enab	Force_rate	Converts the error between the sensed contact wrench and the desired contact wrench of the MERLIN into a rate command.
force_ref.c	Tm_Stool, Fm_Stool, Ts6, Tm0, T60, Ob_mass, Cen_of_grav	Fref_wrtm	Converts the sensed contact forces of the MERLIN to the mrf frame. This module also defines the sensor offset during its first cycle. The outboard weight as well as the offset are removed from the sensor readings within this module.
jr3fts.c	none	Fm_Mwrist Tm_Mwrist	Reads the JR^3 sensor mounted on the wrist of the FREFLEX.
jr3_serial.c	none	Fm_Stool, Tm_Stool	This module uses the serial IOD driver named isio0 on the processor named Control to read the JR^3 sensor mounted on the wrist of the MERLIN.

Table B-1. (cont.)

Module	In Variables	Out Variables	Description
mer_fwd_kin.c	Tbwo, T0b, Theta, Alpha, A, D, Neigh_tmat, T20, T30, T40, T50, T60, Tm0, Tm6	Neigh_tmat, T20, T30, T40, T50, T60, Tm0, Tmwo	Calculates the forward kinematics of the MERLIN.
mer_jacob.c	Neigh_tmat, T20, T30, T40, T50, T60	Mer_jacob	Performs some MERLIN forward kinematics calculations. Also calculates the MERLIN Jacobian of frame {6} with respect to frame {0} expressed in frame {0}.
mer_jnt_pose.c	Theta	Theta, Merlin_enab	Joint position control for the MERLIN. This does not require the use of the FREFLEX, only the keyboard.
mer_jnt_rate.c	Theta, Mer_cmd_mat, Dm_cycle, Merlin_enab	Theta, Mer_cmd_mat, Damp_exo, Ret_to_ctr, Dm_cycle	Joint rate control for the MERLIN. A vertical displacement of the FREFLEX corresponds to a rate command for the chosen MERLIN joint.
mer_pose_ip.c	T0b, Tm6, D, A, Tbwo, Mer_cmd_mat, Tcwo, Tmwo, Theta, Merlin_enab	Theta	Pose control of the MERLIN using inverse position kinematics as derived by Lt. Kurtis Johnson.
merliniod.c	Theta, Merlin_enab	none	This reconfigurable module is intended for use with the new (7/97) MERLIN HSHI serial interface. It writes position commands to the serial interface in terms of encoder ticks. This module uses the serial IOD driver named isio.

Table B-1. (cont.)

Module	In Variables	Out Variables	Description
mrr_forces.c	T60, Mer_jacob, Vel_mer, Rot_vel_mer, Tm0, Tm6, Theta, Merlin_enab, Force_rate	Theta	Calculates theta commands to be sent to the MERLIN using the the resolved rate method of rate control. This module is for use with mer_jacob.c and aux_forces.c. Also can receive force control input for use as Naturally-Transitioning Rate-to-Force Controller. Rot_gain and vel_gain can be set to zero and used in conjuction with force_ctrl.c to operate purely in force control mode.
outbrd_wght.c	Tm_Stool, Fm_Stool	Theta, Ob_mass, Cen_of_grav, Merlin_enab	Calculates the weight outboard of the JR^3 sensor mounted on the wrist of the MERLIN and the center of gravity of the outboard mass with respect to the sensor frame.
pose_diff_mat.c	Qvec_exo, Rmat_exo, Mer_cmd_mat, Merlin_enab	Mer_cmd_mat, Dm_cycle	Generates the grip pose difference matrix of the FREFLEX.
r_exo_pot.c	none	Am_Mjoints, EXO_POTS, Fm_Melbow	Reads the exoskeleton joint potentiometer voltages via the Data Translation DT1401 analog to digital converter board. This module accesses the SAI driver jointpot.c which performs an IODRead() to the device driver dtadc0, an IOD driver for the DT1401.

Table B-1. (cont.)

Module	In Variables	Out Variables	Description
sunenet.c	Am_Mjoints, Qvec_exo, Rmat_exo, Fm_Mwrist, Tm_Mwrist, Gtrq_exo, Dim6_ft, Dtrq_exo, Jacob_exo, Dacs_exo, Ctrq_exo, Fm_Melbow, Elbow_exo, Grip_exo, Grip_dsrd, Elbow_dsrd, EXO_POT, Err_trq_exo, TASK_WALL, FR_XYZ, Mer_cmd_mat, Theta, Fm_Stool, Tm_Stool, TASK_BD_DAT, Total_work	EXO_ELBOW, EXO_TORQ	Sends information to the Sun via the ethernet (UNIX sockets).
work.c	Tm0, Fref_wrtm, WOS_RECORD	Total_work	This calculates the work done by the MERLIN on the environment. This module was used to compare various control methods.